AD-408 250

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

25G-19

[illegible] DATA TRA[illegible]SSION THROUGH [illegible]IMPLE NETWORKS: [illegible] CHARACTE[illegible] [illegible] NODAL ST[illegible]AGE

[illegible] was performed at Lincoln Laboratory, [illegible] Massachusetts Institute of Technology; this work was supported [illegible] the U. S. Air Force under Air Force Contract AF 19(628)[illegible]

LEXINGTON MASSACHUSETTS

Rept. 25 G-17
ESD-TDR-63-76 66

40820 0

ABSTRACT

The case is considered of a single constant rate source of data communicating with a single sink of data, via a set of fault-liable HF links which are connected to relay stations equipped with storage facilities. Methods of transmission are used between nodes which effectively result in a node ceasing to transmit when a link becomes faulty and only resuming when a link returns to its normal condition. Thus data will queue at a node when the outgoing link is faulty, and all data reaching the sink is error-free. The network has been simulated on a digital computer using the recorded link behavior of two existing HF.RTTY links. The probability distributions have been obtained for the queues that arise at the nodes and for the total delay involved in transmitting data from the source to the sink.

I. Introduction

Some problems arising from transmitting data through fault-liable links connected in series are considered.

A source of data is communicating with a sink of data by means of a set of communication relay stations (see Fig. 1). The communication stations (nodes) are connected together by two-way links, the forward channel being utilized for data and the backward channel for feedback signals as described under single-link operation. All nodes have reception, transmission, and storage facilities. The essential components of a relay node are shown in Fig. 2.

II. Data Format

The data is divided into equal length segments. Redundant data is added for purposes of error detection in accordance (in this case) with the Bose-Chaudhuri code. The resulting segment of data is called a block. All blocks dealt with in this report contain 255 bits.

III. Data Source

A constant rate data source is assumed in order to separate queueing effects due to link behavior from queueing effects due to source statistics. If, in a given situation, data does not arise in a uniform manner, then additional storage may be required at the source node to smooth out the fluctuations in the source statistics.

IV. Single-Link Operation

The basic method of transmitting data through a single link is described in a paper by Reiffen, Schmidt, and Yudkin.[1]

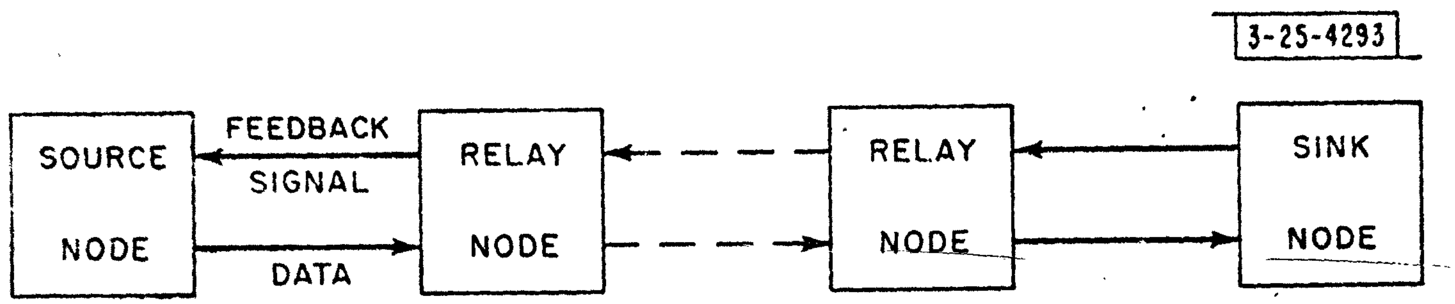

Fig. 1. Network Constituents

In the present report data is flowing in one direction alone--from the source node to the sink node. The reverse direction is used only for feedback--to inform the source as to the correctness of the data received at the sink. In practice this is a completely symmetric system with data and feedback signals flowing in each direction (or channel).

If the sink detects an error in a received block, then the block is ignored, and the sink sends back to the source via the feedback channel a request for the retransmission of this block. This process of error detection and repetition may continue indefinitely until the block is received correctly. The net effect of this system is to make the source appear to have stopped sending data while the link is in a faulty condition.

Due to the finite time taken to transmit blocks over a link, this simple picture has to be amended slightly. On reception of an erroneous block, the sink places a request for a retransmission in the block which is currently leaving the sink for the source via the feedback channel. By the time the source receives this request, it is already transmitting the block subsequent (in the go-back-two system) to the erroneous block. The system logic is such that this subsequent block is erased as soon as it reaches the sink, and the source will retransmit both the erroneous block and the following block (whether or not the latter is erroneous). If the transmission delay in the link is such that the source receives a feedback signal relating to the first block transmitted while it is transmitting the n^{th} block, then we have a go-back-n system in which the (n-1) blocks following an erroneous block will be erased at the sink, n blocks being

retransmitted from the source. In order for the source to be able to repeat data, it must have storage facilities for a few blocks. (This number is dependent on n.) This storage is not included in the node storage requirements which are noted later as being required by link outages.

The possibilities of errors occurring in the feedback channel must be considered. In this report the source will assume that any erroneous blocks received on the feedback channel did, in fact, contain a request for retransmission and will repeat the block to which this request would refer. However, there is another method of transmission (Error correction by requested retransmission. See Appendix I) where the source continues to transmit while the feedback channel is faulty but simultaneously sends the data to local storage as well. When the feedback channel resumes normal operation, the sink will inform the source which, if any, blocks should be retransmitted; and the local storage is cleared of all blocks that do not have to be retransmitted. This method of transmission is mentioned because although it does not significantly alter delay, it does increase the effective data-carrying capacity by making use of storage. In the case of links with a channel reliability of 0.90 (see Sec. V), the increase is about 12 percent.

V. Channel Reliability Figure

The reliability of one of the two channels of a link is denoted by ρ and is defined by the equation

$$\rho = \frac{\text{Number of blocks received correctly from the channel}}{\text{Total number of blocks received from the channel}}$$

ρ is referred to in the sequel as the channel-reliability figure. This ratio was obtained for the total number of blocks transmitted during a channel test and thus measures the average reliability of a channel.

VI. Network Utilization Figure

Network utilization figure denoted by λ is defined by

$$\lambda = \frac{\text{Source rate}}{\text{Effective network rate}},$$

where the rates are expressed in blocks per unit time, and the effective network rate is the average number of error-free blocks that are transmitted from source to sink per unit time. This is a useful measure of the degree to which a network is being used. However, when comparing two methods of transmission (Sec. V) over the same network, a better measure is the ratio

$$\mu = \frac{\text{source rate}}{\text{channel rate}}$$

where the channel rate is the total number of blocks (error-free and erroneous) that can be transmitted through the network per unit of time. It may be noted that $\mu = \rho\lambda$.

VII. Node Storage

In the simulation infinite node storage was assumed. However, in all cases for which the network utilization figure was less than one, the queues were bounded, and hence in practice finite storage would suffice.

An alternative system arrangement is possible employing relay stations having regeneration and/or amplification, but without storage facilities. These would send on data to the next node immediately on

receiving it from the previous node. If this is done (i.e., if there is no storage at intermediate nodes), then the data-handling capacity is reduced and delay is increased. As one example, consider a cascade of five nodes and a channel reliability of 0.90. With storage, data can be transmitted at a rate more than three times the rate data can be transmitted if no storage were used. In the case of more reliable links, storage may not significantly affect the data-handling capacity. With a link reliability of 0.986 and a five-node network, the provision of storage at intermediate nodes gives less than one percent increase in average data-handling capacity. Even if storage is omitted at intermediate nodes, storage still will be required at the source node.

VIII. Single Link Behavior

The behavior of a single HF channel using a 255-bit block length (Bose-Chaudhuri code) was obtained by A. B. Fontaine[2] (see Appendix II). The channel behavior may be characterized by a sequence of 1's and 0's, where one indicates a block without errors and zero indicates a block with errors. This data is summarized in Figs. 3 and 4 for the New York - Bermuda and New York - Johannesburg links. A run refers to a series of successive 255-bit blocks, each of which has one or more errors. The existence of some long error runs typifies the behavior of HF links. The Bermuda link is assumed to be typical of "good" links ($\rho = 0.986$). Some idea of how the channel behaves may be gained from Fig. 5 and Fig. 6 where the incidence of error blocks per consecutive 1000 blocks is shown as a function of time during the test. These samples refer to one direction

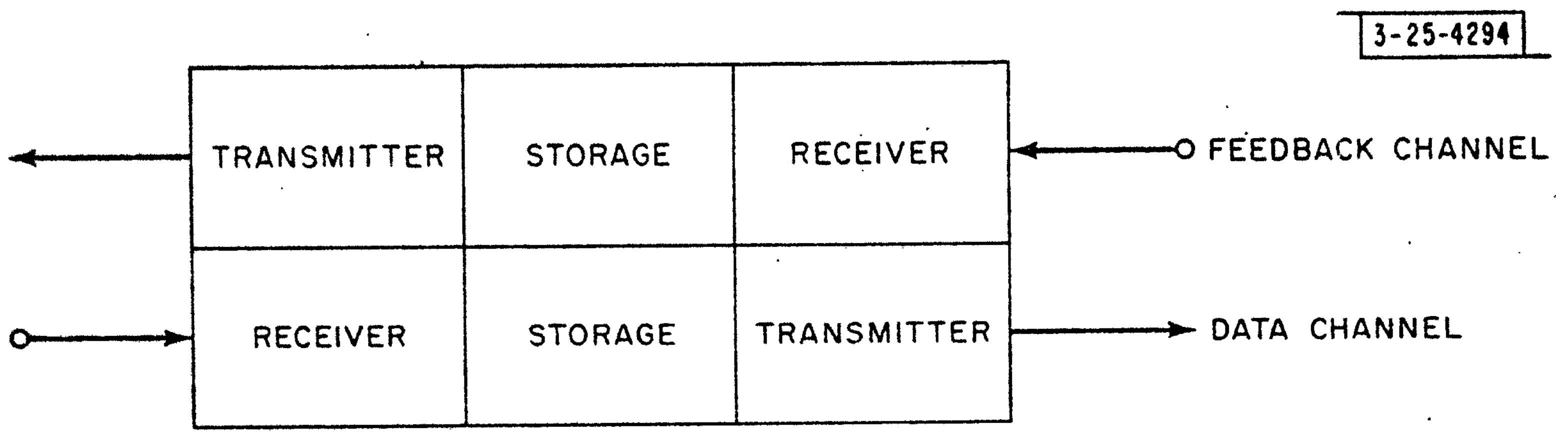

Fig. 2 Relay Node Components

of the link (a channel), but in the normal go-back-two method, the feedback channel interacts on the forward channel; while the feedback channel is faulty, the source is effectively cut off. To obtain the behavior of the complete link, it has been assumed that the feedback channel behaves in the same way, but is shifted in time with respect to the forward channel. Thus the same samples were used for the feedback channel but shifted by one-third the length of the total data. This time shift was chosen arbitrarily since no data is presently available regarding the behavior of the two channels of a link. It may be noted that the data-carrying capacity of the link is a maximum if there is no time shift between the two channels. One simulation was performed with each assumption, and the results are given in Sec. XIII.

IX. Network Behavior

First, it is assumed that for a given simulation all links are of the same type (either "good" or "poor" links). Second, all links are assumed uncorrelated with respect to each other. The effects of this latter assumption are investigated in Sec. XIV.

X. Formation of Queues

The ground rules for this study include the following premises:

1. no data can be destroyed, and
2. the data source cannot be turned off.

Thus when a link is faulty, data will queue in the preceding node. This model is a non-priority system in which, when the link returns to normal, the first block to have reached the node will be the first to leave. With a

utilization figure less than one, the data blocks are interspersed with periods when the source is not transmitting data. Consider the output of a node whose outgoing link suffers a long noise burst (with consequent queue formation) followed by a noiseless period. During the noise burst there is no input to the next node. After the noise burst the queue empties into the next node at the maximum possible rate with no interspersed non-data periods. The net effect of noise bursts in this situation is to bunch the data blocks together.

XI. Delay

The blocks are delayed by the amount of time spent in queue storage. Delay is measured as the difference between the time the block actually takes to reach the sink and the time it would take to reach the sink if the links were perfect.

XII. Results of the Simulation

Frequency distributions of queues were found for all nodes, and the frequency distribution of delay for each network was found. Networks with different numbers of nodes were simulated, and different network utilization figures employed.

XIII. Effect of Channel Correlation

In Fig. 7 the effects of correlation between the feedback and data channels are shown. In the case of negative correlation* the sample of feedback channel behavior was identical to that of the data channel but shifted in time by approximately one third of the total duration of the test.

* Negative correlation is used in the sense that successive links are in different states at the same time.

In the case of positive correlation this time shift was zero. It will be noted that for short- and medium-length queues the frequency with which queues occurred was almost double in the case where the channels were negatively correlated. However, the maximum-length queues observed were almost the same, since these maximum-length queues are due to the widely separated long error runs.

XIV. Effect of Link-to-Link Correlation

Figures 8 and 9 show the effect of correlation between links in a two-link network on the queues that arise at node 2. If there is complete positive correlation between all links (i.e., if all links are in the same state at the same time, either transmitting or not transmitting at the same time), then no queue will arise at node 2. Figure 8 shows the queue distributions at node 2 for a low-network utilization figure for various time shifts between the links. An increasing time shift causes an appreciable increase in frequency of non-zero queues up to a time shift of about 400 blocks. This indicates that the small individual error runs which cause queues at low-utilization figures have a nearly maximum negative correlation if there is a time shift between links of 400 blocks. However, a high-network utilization figure (Fig. 9) positive correlation is still well in evidence for a time shift of 400 blocks. This may be attributed to the fact that the queue due to one error burst has not collapsed by the time the next burst occurs; in effect, a smearing of the error bursts takes place. These long smeared error bursts are still positively correlated for time shifts between links of 400 blocks. Maximum negative correlation was achieved for a time shift of about 25,000 blocks. In the remainder of this report, unless the contrary is stated, a time shift of 20,000 blocks between links has been assumed.

Some evidence has been obtained recently[3] that inter-error intervals may be independent and have a simple statistical relationship. If further investigation substantiates this, then it will be possible to use the Monte Carlo method to obtain the behavior of each link assuming that the links are statistically independent and, therefore, uncorrelated in the strict statistical sense.

XV. Queue Equilibrium

Two conditions have to be fulfilled for queues to reach equilibrium:

1. A sufficient number of blocks should be sent through the system so that the initial transients are eliminated. Queue distributions were obtained after sending through 128,000 blocks (i.e., the complete channel sample was used once), and then the queue distributions were obtained by sending through 256,000 blocks (i.e., the complete sample was used twice). The two sets of distributions differed by less than 1 percent, indicating that transients had little effect after 128,000 blocks were sent through the network. In all the simulations of this report, data were therefore sent through until the complete channel sequence available had been used once.
2. The average rate of the data source must be less than or equal to the average rate at which the network can transmit data, i.e., $\lambda \leq 1$.

XVI. Network with more Reliable Links ($\rho = 0.986$)

A. Queue Behavior

The case of ten links connected in series was taken, and the queue distributions for all nodes were found. Figures 10 through 13 show the distributions for nodes 1, 5, and 10 for four different values of network utilization. Figure 14 is for a time shift of 400 blocks between links. The following points may be observed:

1. For a given network utilization figure the maximum storage requirement is approximately the same for all nodes.
2. For values of the network utilization figure between 0.2 and 0.9, the maximum queue length for any node increases exponentially with increasing network utilization figure. (See Fig. 15).
3. For a network utilization figure less than one, queues tend to occur with greater frequency for nodes nearer the sink. This may be explained as follows:

   The data enters the first node at an even rate, but as noted in Sec. X, it tends to be bunched on entering the second node due to burst noise interruptions of the first link. There is a probability that this concentrated stream of data will enter the second node while the outgoing link from this node is experiencing

a noise burst, (see Sec. XIV). From the decreasing nature of the frequency density function of the queues, it follows that the frequency distribution curve for node 2 will be above that of node 1 and so on for the other nodes. To state the reasoning differently: As the data proceeds through the cascade of nodes, it tends to become more and more bunched due to noise bursts experienced in successive links, with the consequent formation of larger queues.

4. In Sec. XIV it was observed that, for a time shift between channels of 400 blocks, positive correlation effects not in evidence at low-network utilization figures become marked for high-network utilization figures. It may be seen that for low-network utilization figures, the queue distributions assuming a time shift between links of 400 blocks are very similar to those obtained for a time shift of 20,000 blocks. However, for a high-network utilization figure and a time shift of 400 blocks, the order of the curves is inverted (compare Fig. 12 and Fig. 14) due to the correlation of the long smeared error bursts in the incoming link of node 2 and with those in the outgoing link with a consequent reduction of queues at node 2 in comparison with those at node 1.

5. Figure 13 shows an inversion of the order of the curves. However, in this case the queues are not in equilibrium.

B. Delay Behavior

Networks containing 1, 5, and 10 links were simulated. Regarding the delay experienced in receiving the data at the sink (Figs. 16 through 19), it will be seen that the maximum delay experienced is not strongly dependent on the number of nodes in the network for low- and medium-network utilization figures. For example, Figs. 16 and 17 show that the maximum delays experienced were approximately the same; the main effect of cascading nodes is to increase the frequency of non-zero delays. For high-network utilization figures (e.g., 0.9, see Fig. 18) maximum delay is doubled for a five-fold increase in the number of links in the network; also, the average delay is increased by a factor of five. Again, as for the individual node queues, the maximum delay experienced is an almost exponential function of network utilization figure, increasing rapidly as this figure approaches one. The distributions of Fig. 19 do not, of course, converge as more data are sent through the networks.

XVII. Networks of Less Reliable Links ($\rho = 0.90$)

Networks containing 10 links, 5 links, and 1 link were simulated.

A. Queue Behavior

The results of the simulation are shown in Figs. 20 through 22. The trends are the same as noted under the queue behavior of the more reliable links.

1. For low values of the network utilization figure, the maximum queue length for a given node is approximately proportional to the network utilization figure. It is interesting to note

that the large increase of storage requirements with high utilization figures is not as pronounced in this case as it is in the case of networks with more reliable links.

Placing $\Delta = \dfrac{\text{Percent increase in maximum queue}}{\text{Percent increase in network utilization figure}}$,

the following table may be compiled from the graphs:

	Channel Reliability	Δ
Node 1	0.98	14
Node 10	0.98	14
Node 1	0.90	2.6
Node 10	0.90	2.6

The figures are calculated for the maximum and minimum network utilization figures employed in the simulation. If $\Delta = 1$, storage requirements are directly proportional to the network utilization figure. Thus for less reliable links there is a low incremental cost (in terms of storage) to running the system at high data rates. This is an important result, since it is in this case (less reliable links) that, as pointed out previously, the provision of storage increases the effective data-handling capacity of the system considerably.

2. In general, queues tend to occur with greater frequency for nodes nearer the sink; but for a network utilization figure of 0.9, the queue distribution curves for node 5 and node 10 are in reverse order due to positive correlation between smeared noise bursts in one link and those in the next link.

B. Delay Behavior

Networks containing 1, 5, and 10 links were simulated. The results are shown in Figs. 23 through 25. Here again, maximum delay is not a strong function of the number of nodes in a network. The most noticeable effect of adding more links is that the frequencies of non-zero queues are increased.

XVIII. Comparison of Storage Requirements

On the basis that no data can be destroyed, the storage facilities will have to cope with the maximum queues that may arise. It is interesting to compare the storage requirements of two communication systems, containing the same number of nodes (11), for the same network utilization figure ($\lambda = 0.9$). The first system contains more reliable links ($\rho = 0.986$) and the second system less reliable links ($\rho = 0.90$). For the first system, storage for 300 blocks is required at node 1 and at node 10. For the second system, the storage requirements are 1000 blocks at both nodes. The network utilization figure was defined in a way to facilitate the comparison between two systems with links of different reliability and a less reliable system. This has meant calculating the average reliability of the systems, but recent work indicates that there may be violent fluctuations in the averages calculated over finite samples and that the average over an infinite sample may be infinite.

XIX. Summary

This report has endeavored to give practical answers to the problems involved in sending data through a set of HF RTTY links in series. It has

been shown that the effective data transmission rate may be greatly increased (in the case of "less reliable links") by providing storage at the relay stations (nodes). Storage requirements were found to increase but moderately as the source rate was increased up to 90 percent of the maximum possible rate. The storage requirements have been found and also the total delay experienced by data for several input data rates. Furthermore, it has been shown that adding more links causes only small increases in the maximum delay experienced.

jmc

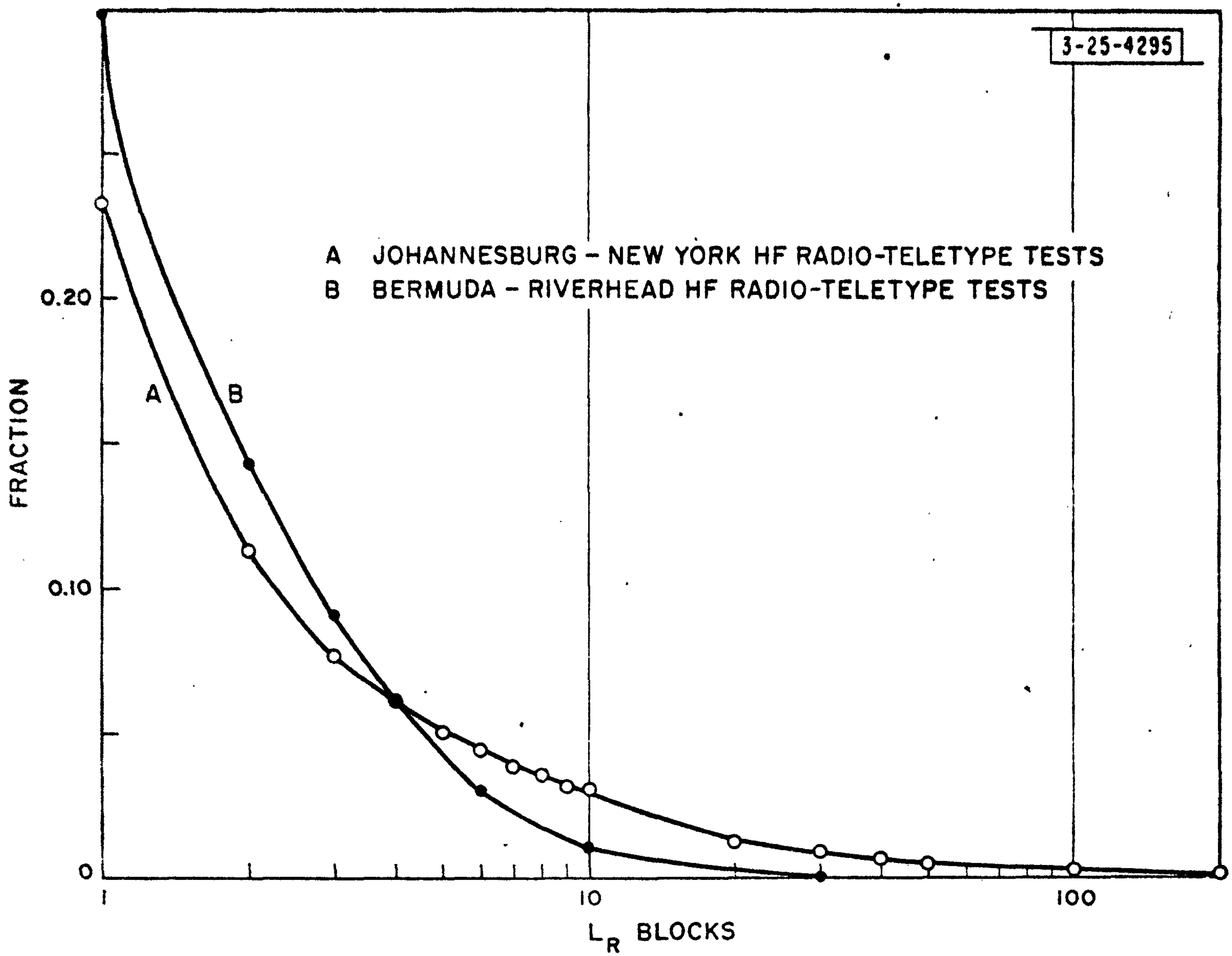

Fig. 3 Fraction of Runs of Consecutive Code Blocks With Errors for Which the Length of the Run is Greater than L_R

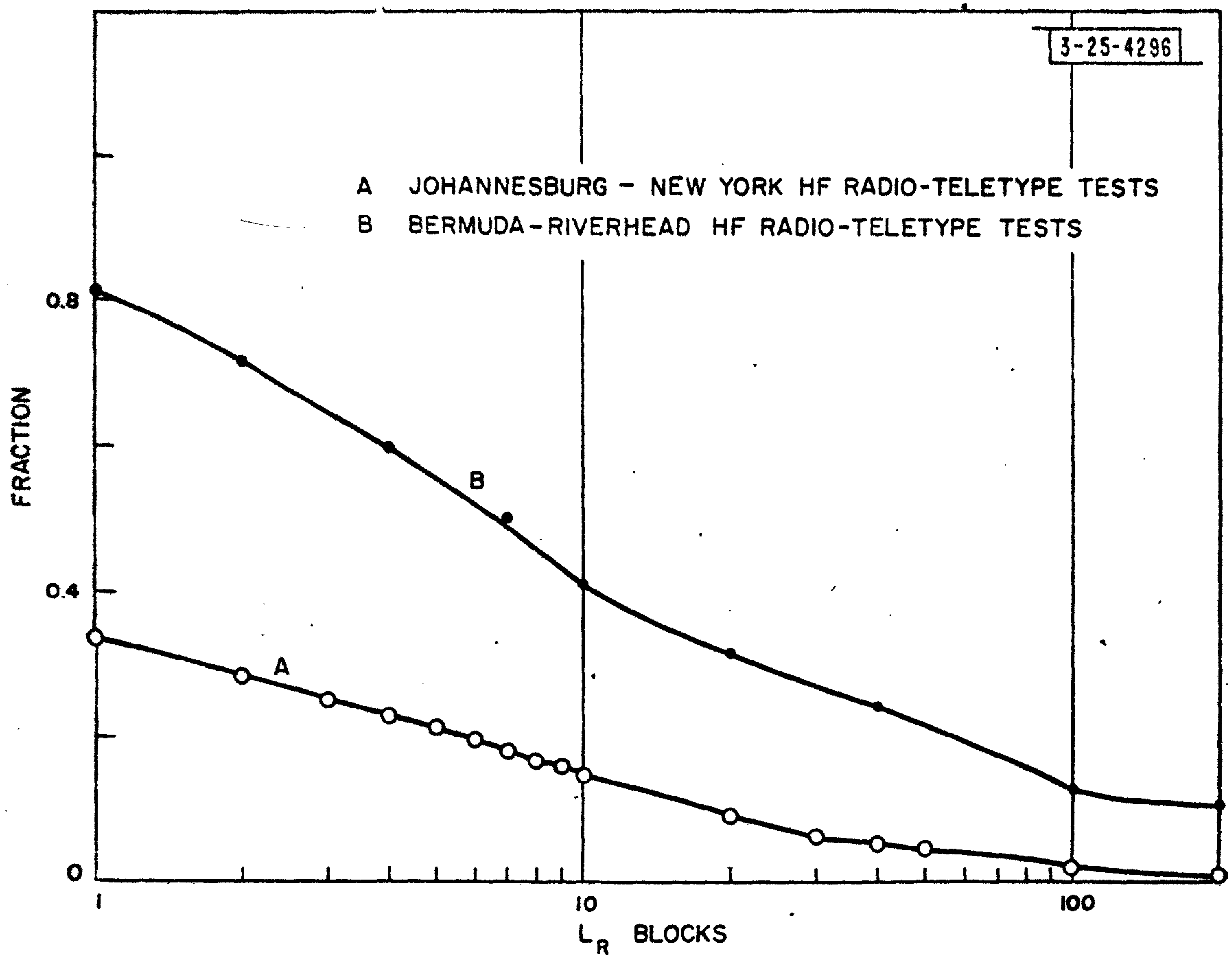

Fig. 4 Fraction of Runs of Consecutive Code Blocks Without Errors for Which the Length of the Run is Greater than L_R

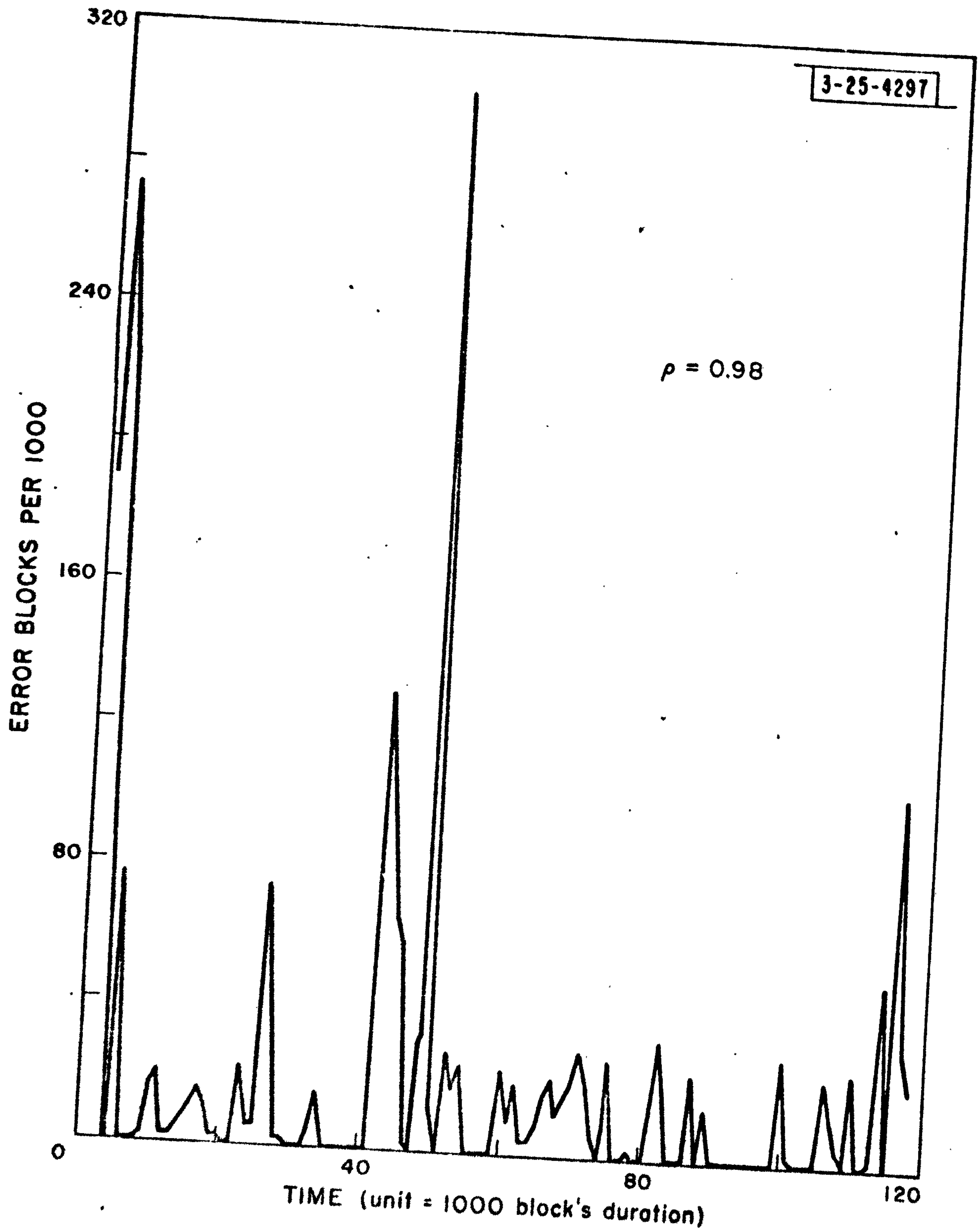

Fig. 5 Error Density as a Function of Time. Bermuda Link

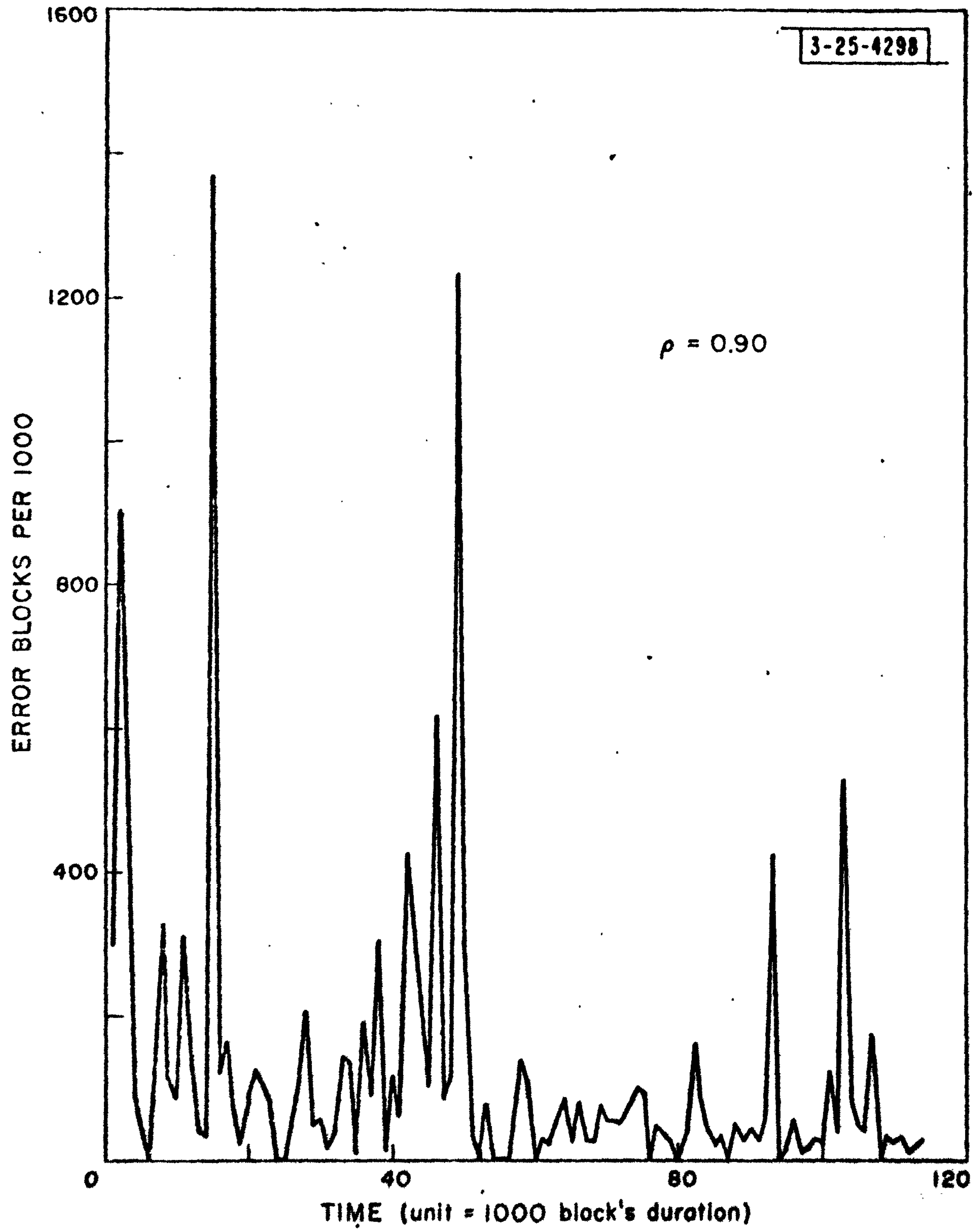

Fig. 6 Error Density as a Function of Time. Johannesburg Link

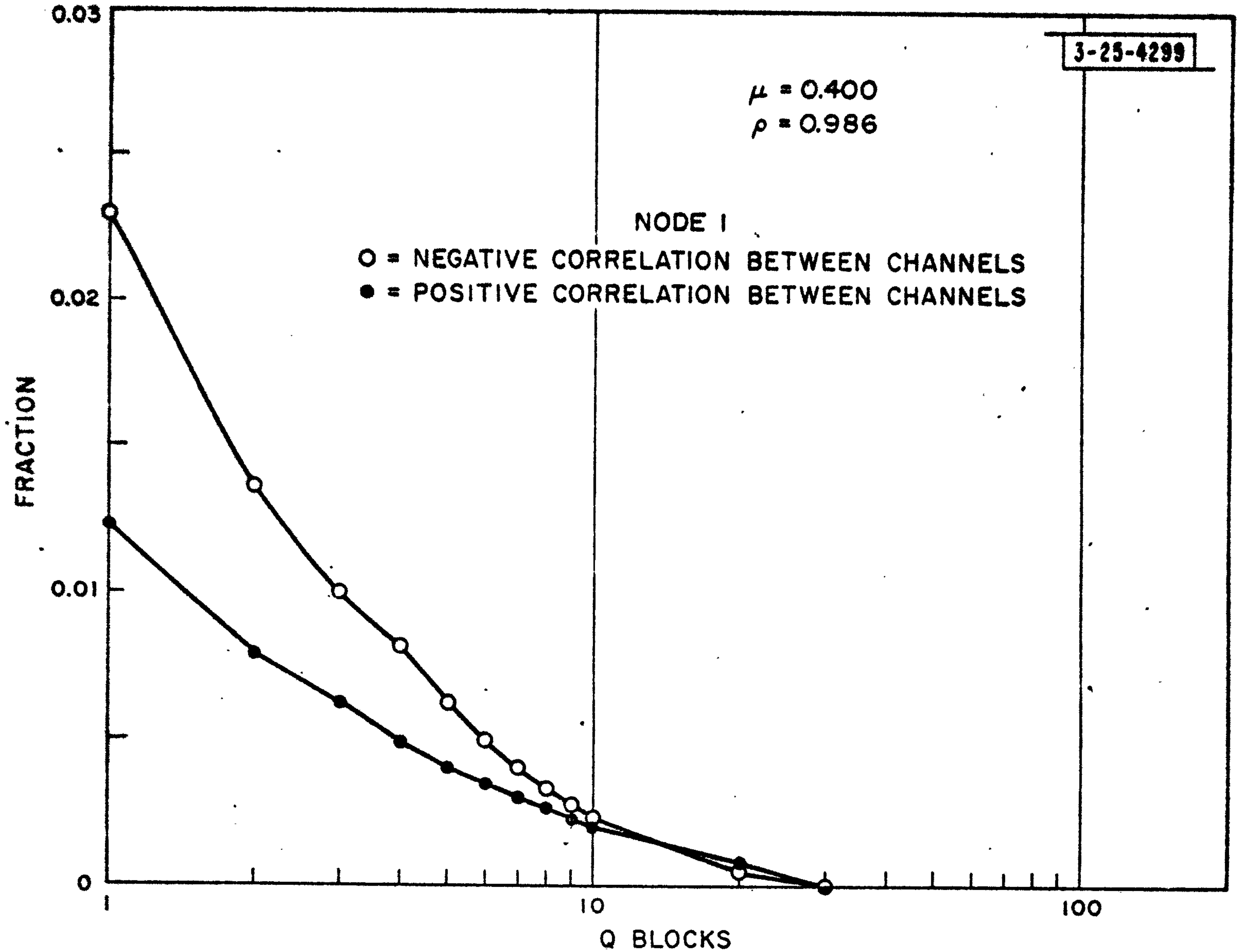

Fig. 7 Effect of Channel to Channel Correlation. Fraction of Queues Containing More than Q Blocks for Node 1

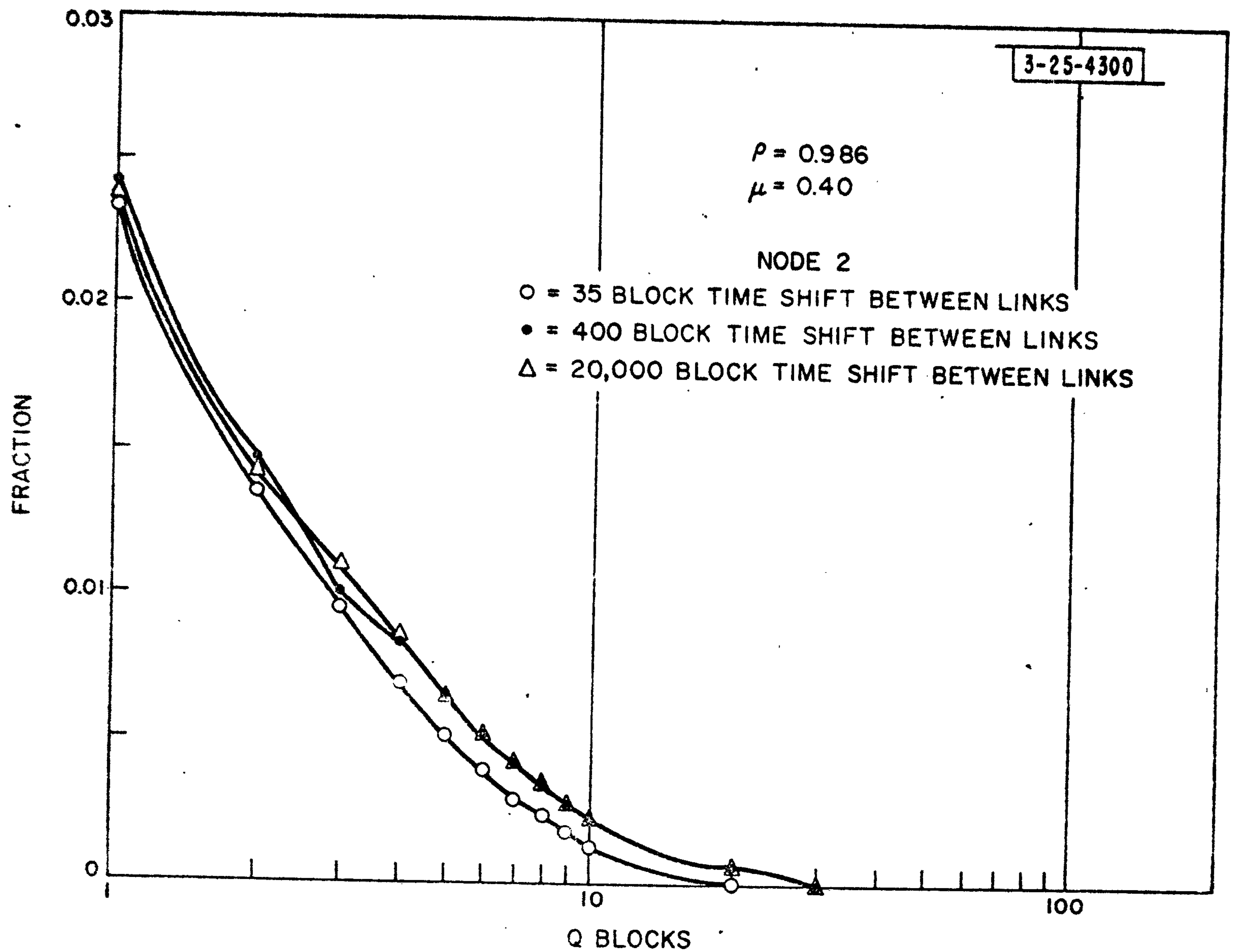

Fig. 8 Effects of Link to Link Correlation. Fraction of Queues Containing More than Q Blocks

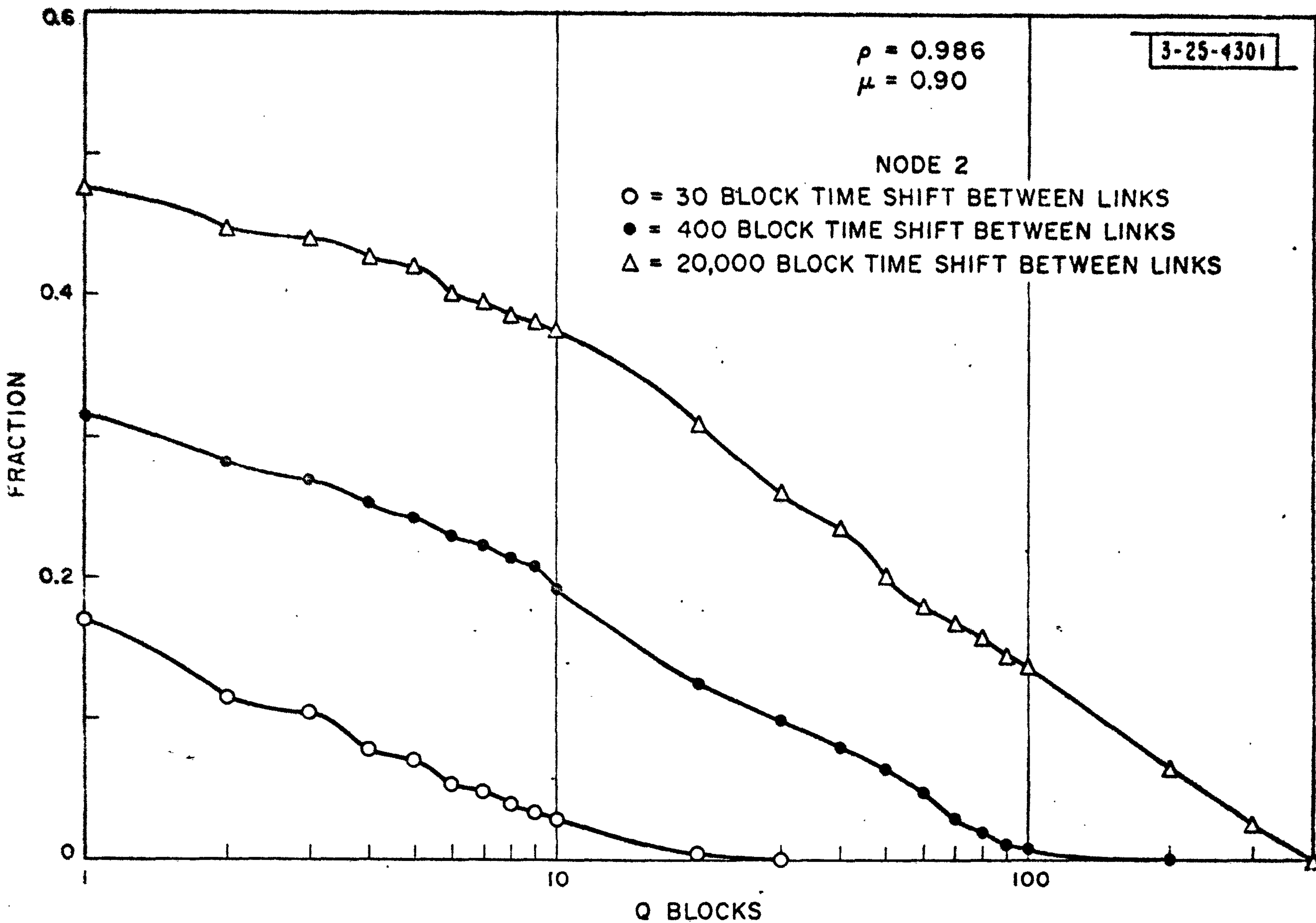

Fig. 9 Effects of Link to Link Correlation. Fraction of Queues Containing More than Q Blocks

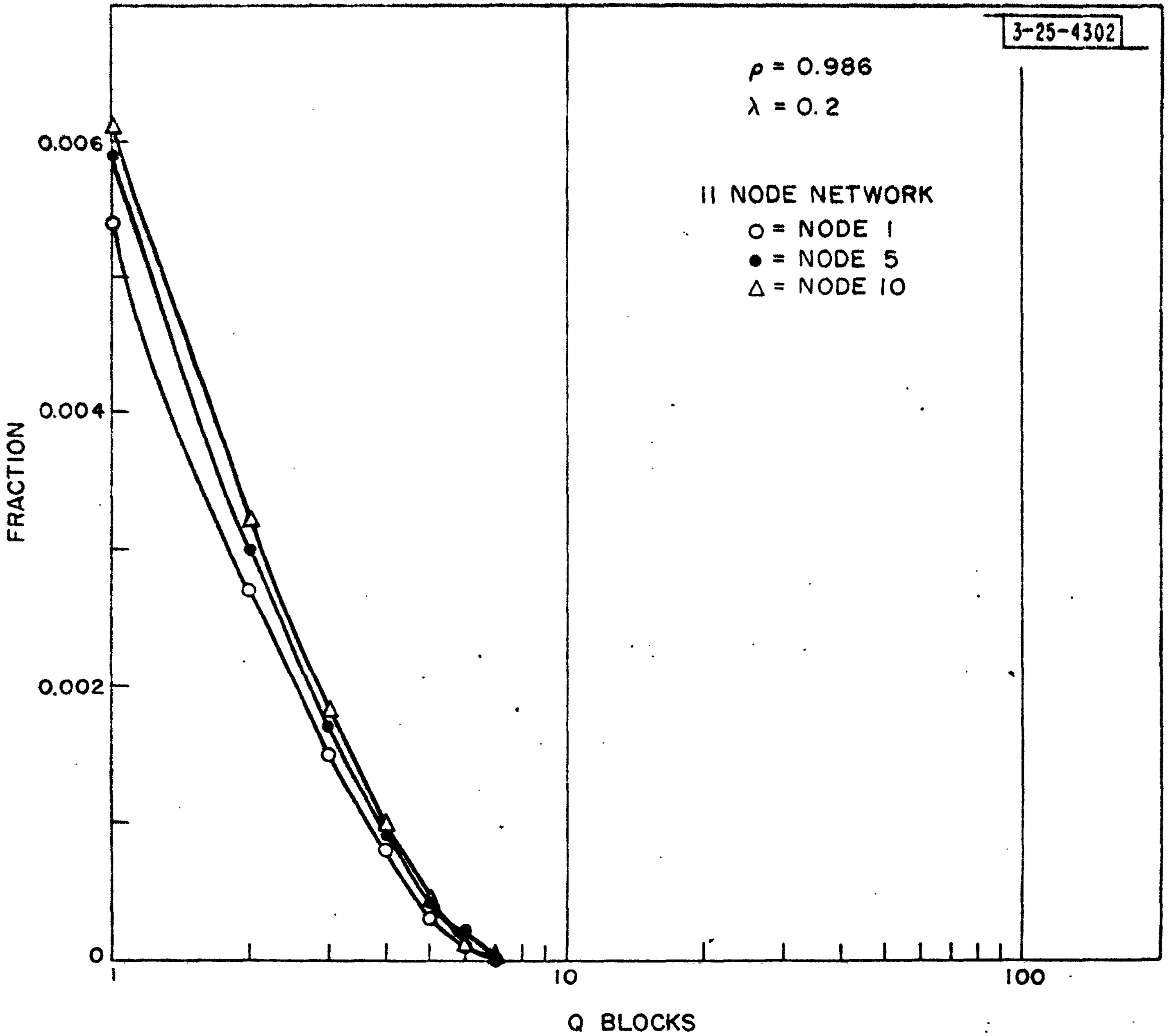

Fig. 10 Fraction of Queues Containing More than Q Blocks

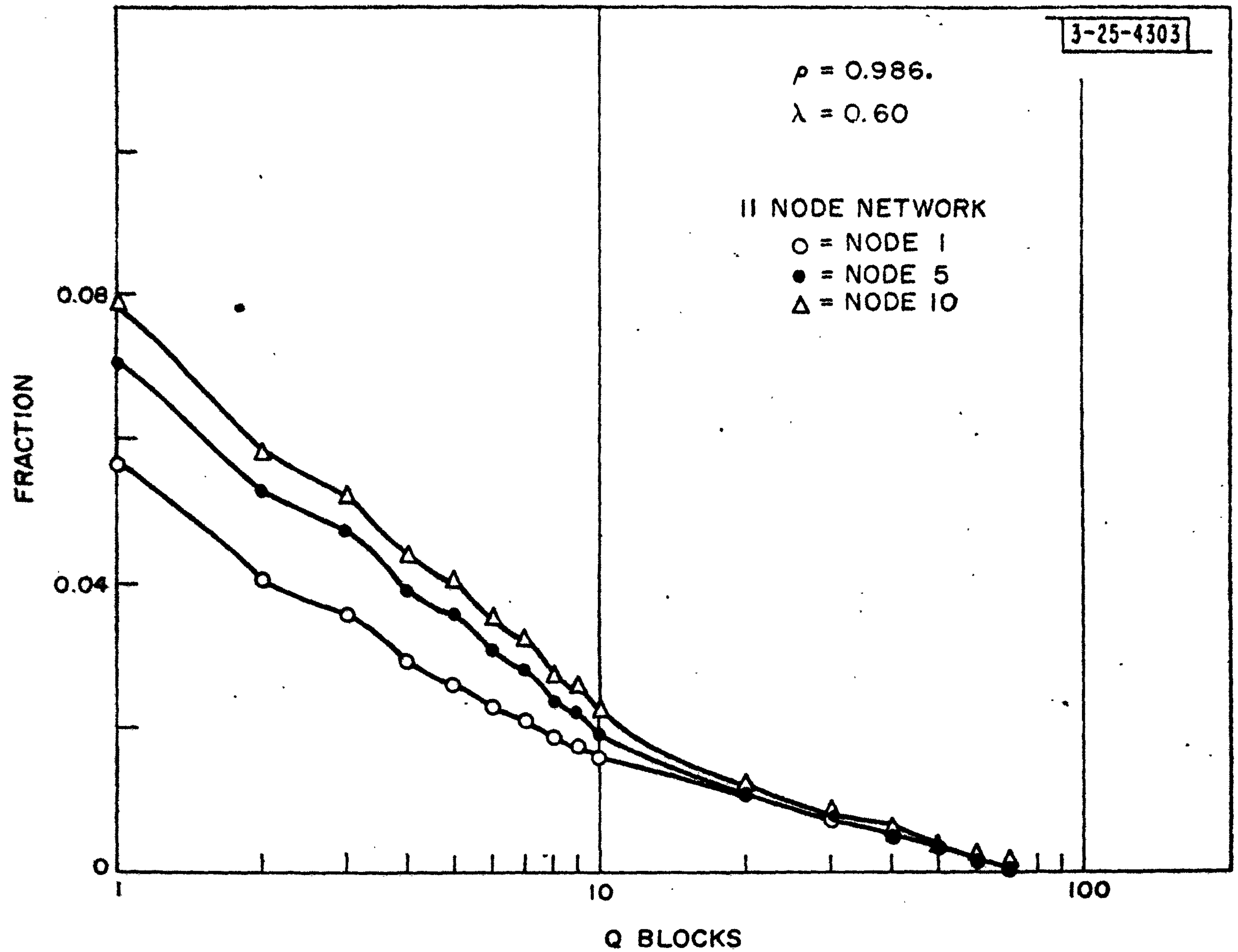

Fig. 11 Fraction of Queues Containing More than Q Blocks

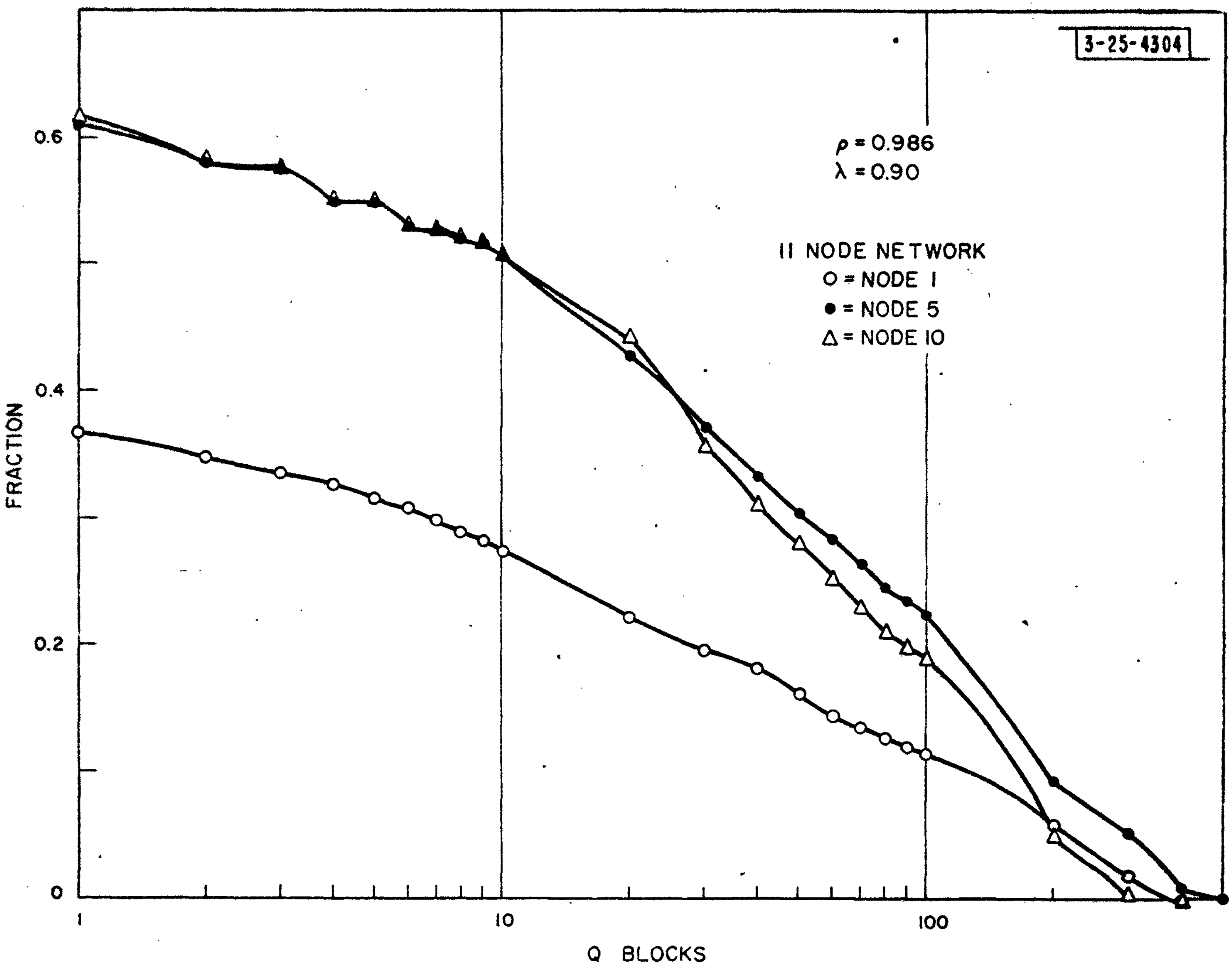

Fig. 12 Fraction of Queues Containing More than Q Blocks

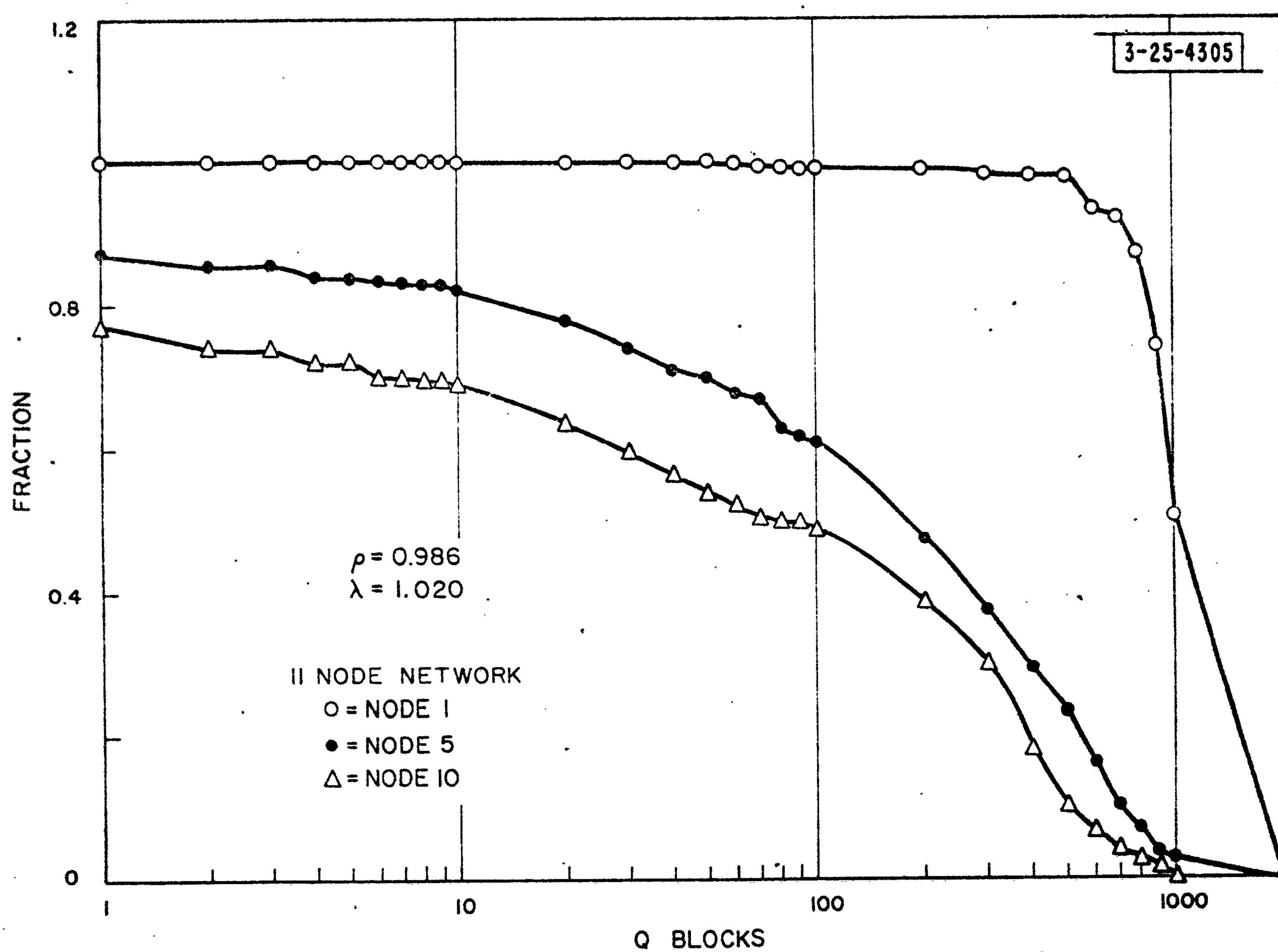

Fig. 13 Fraction of Queues Containing More than Q Blocks

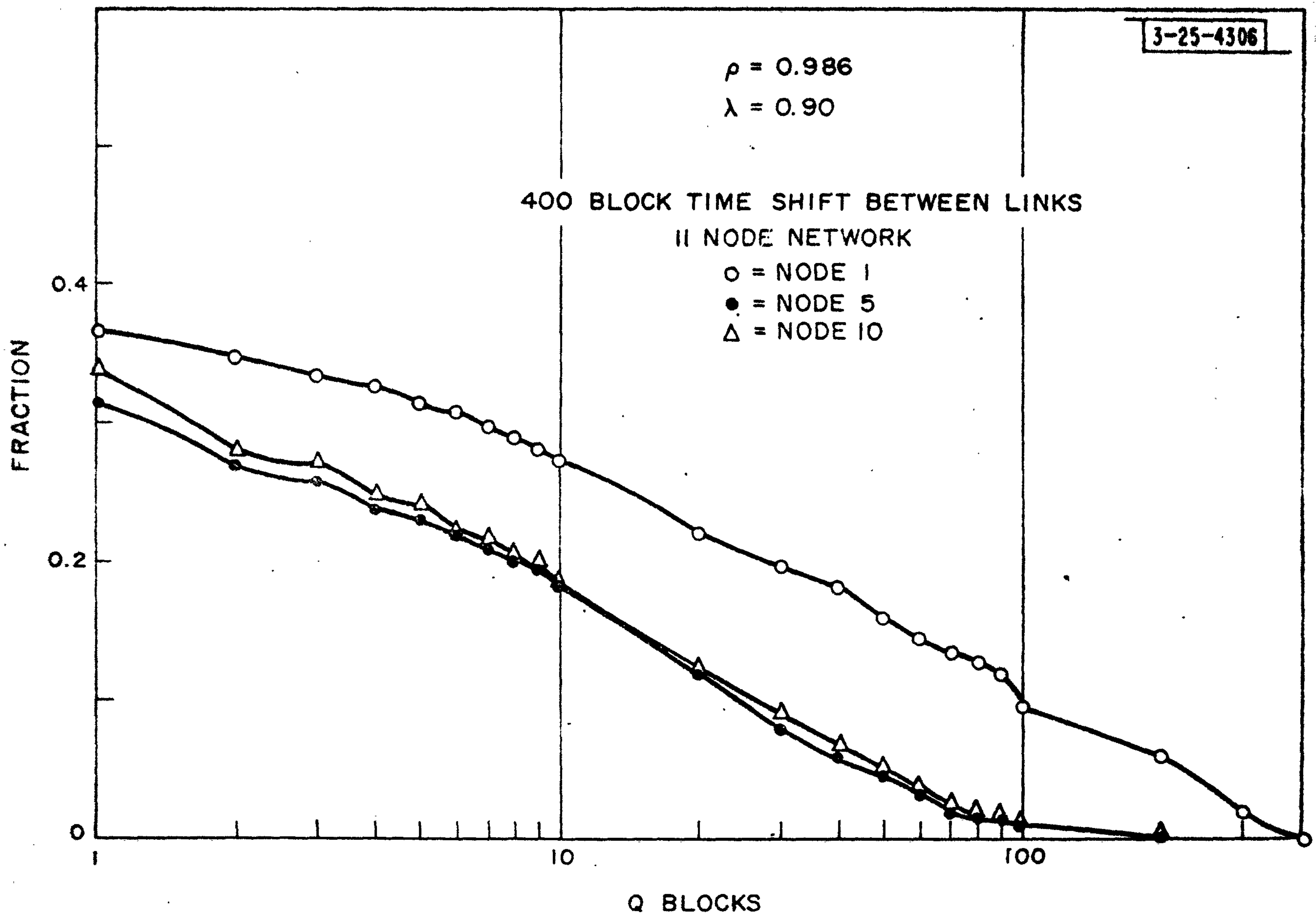

Fig. 14 Fraction of Queues Containing More than Q Blocks

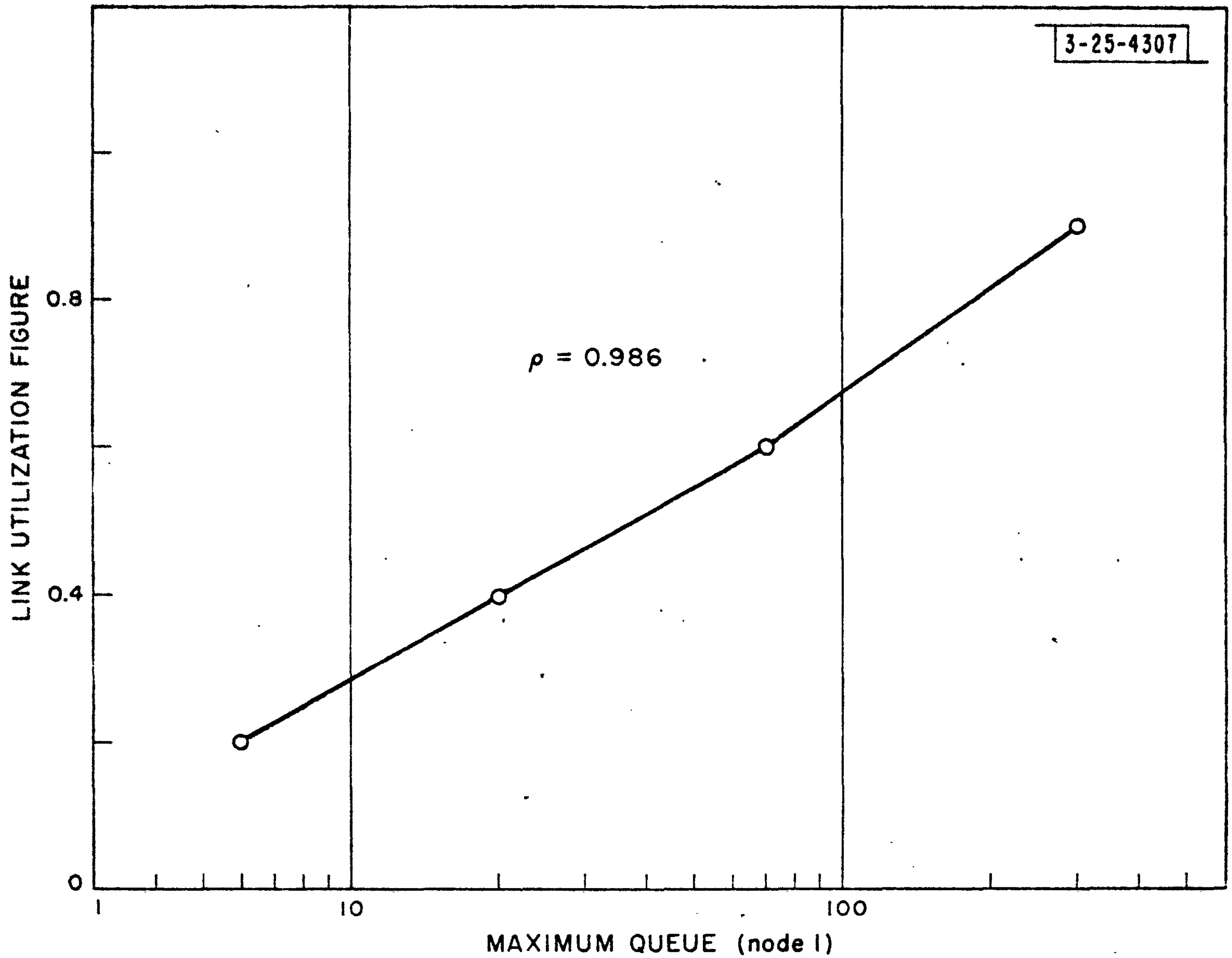

Fig. 15 Maximum Queues as a Function of Link Utilization Figure

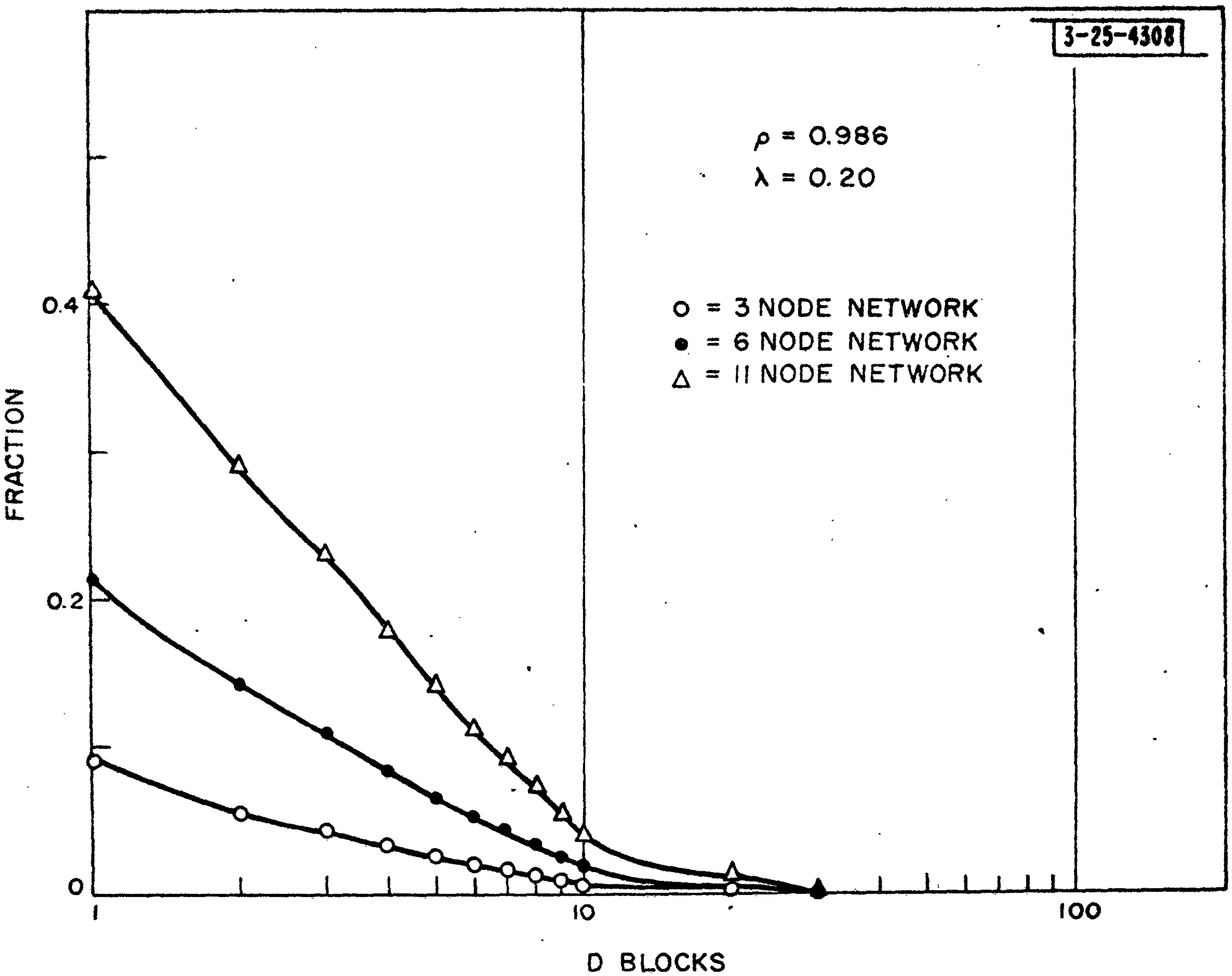

Fig. 16 Fraction of Delays Greater than D Blocks Duration

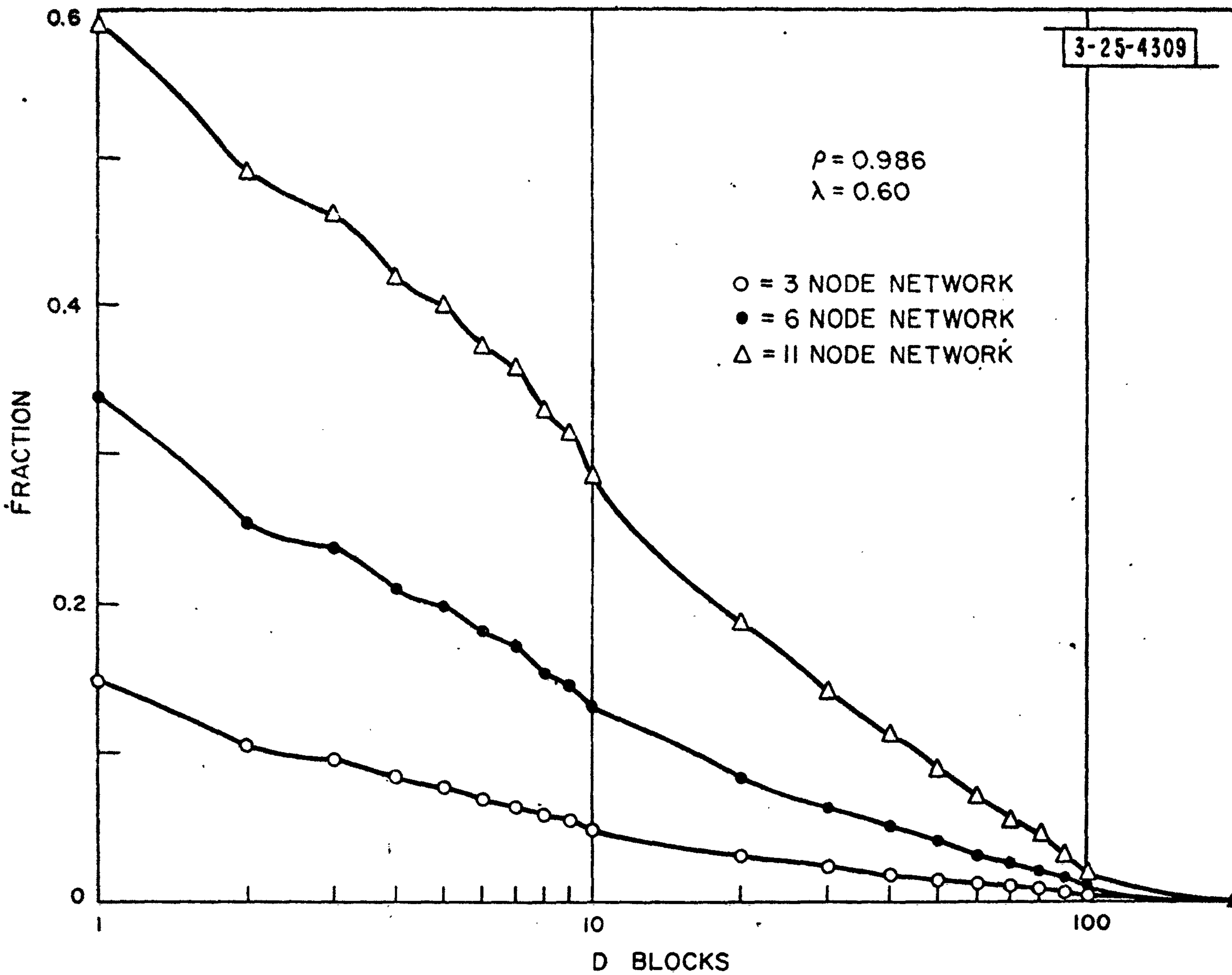

Fig. 17 Fraction of Delays Greater than D Blocks Duration

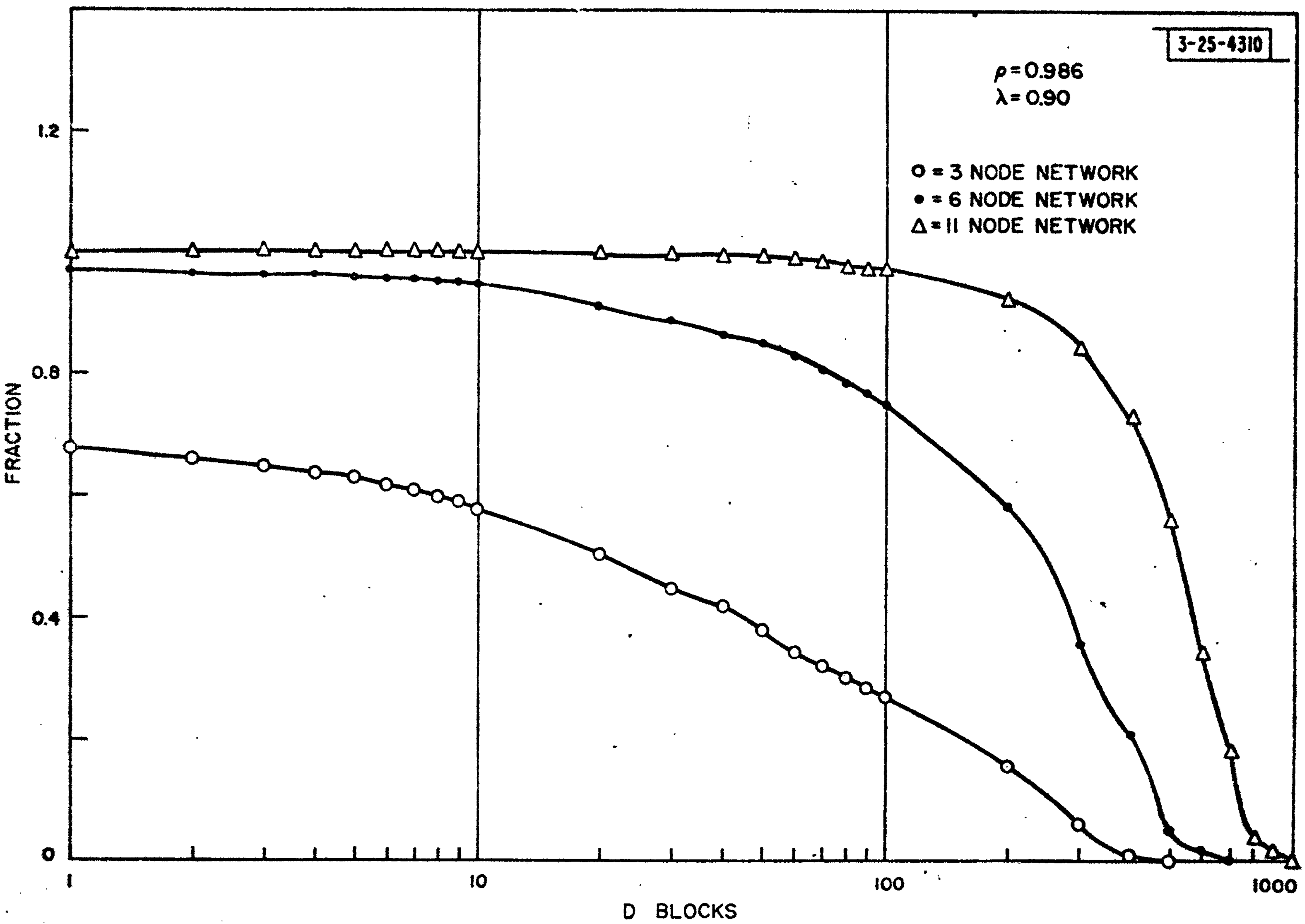

Fig. 18 Fraction of Delays Greater than D Blocks Duration

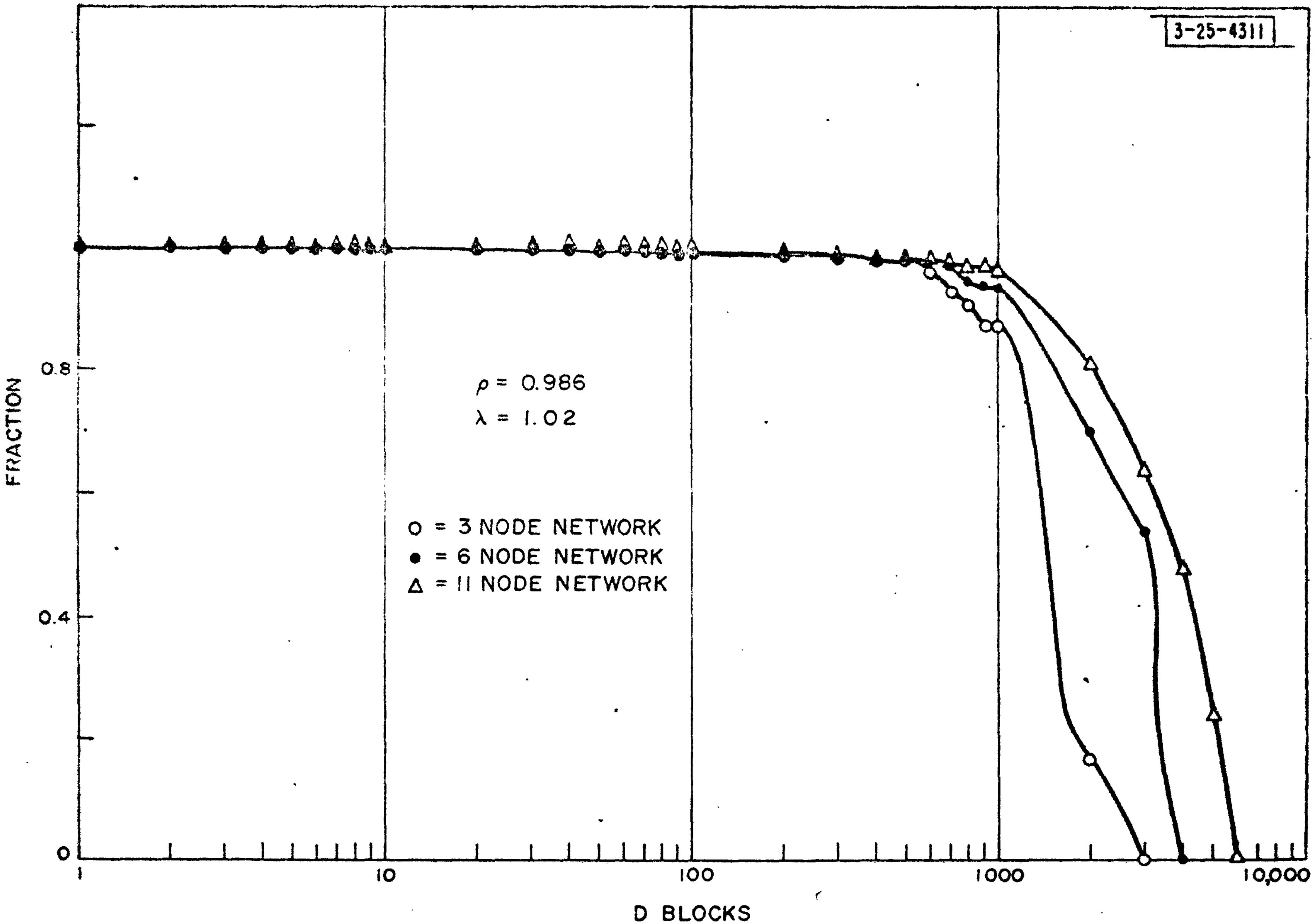

Fig. 19 Fraction of Delays Greater than D Blocks Duration

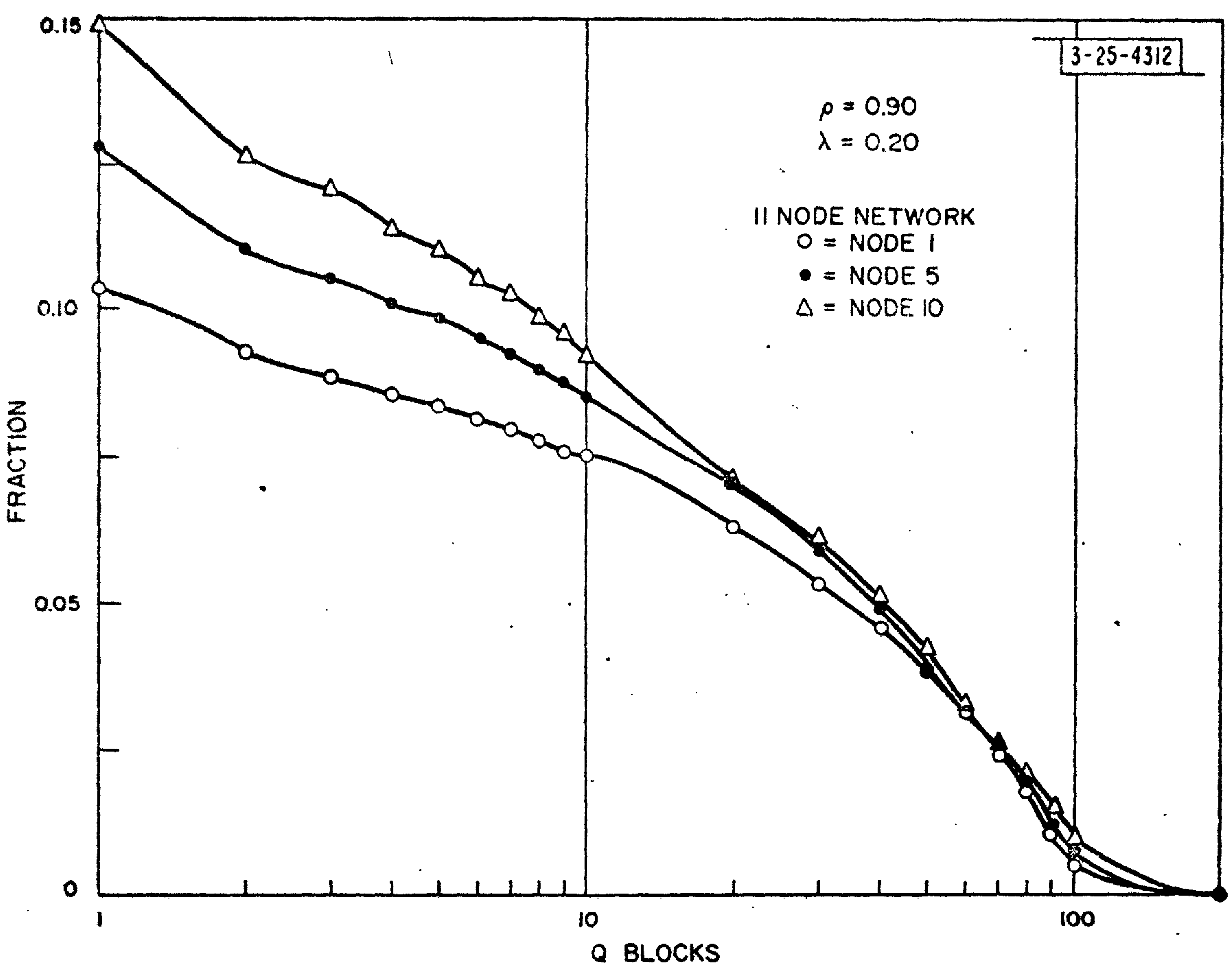

Fig. 20 Fraction of Queues Containing More than Q Blocks

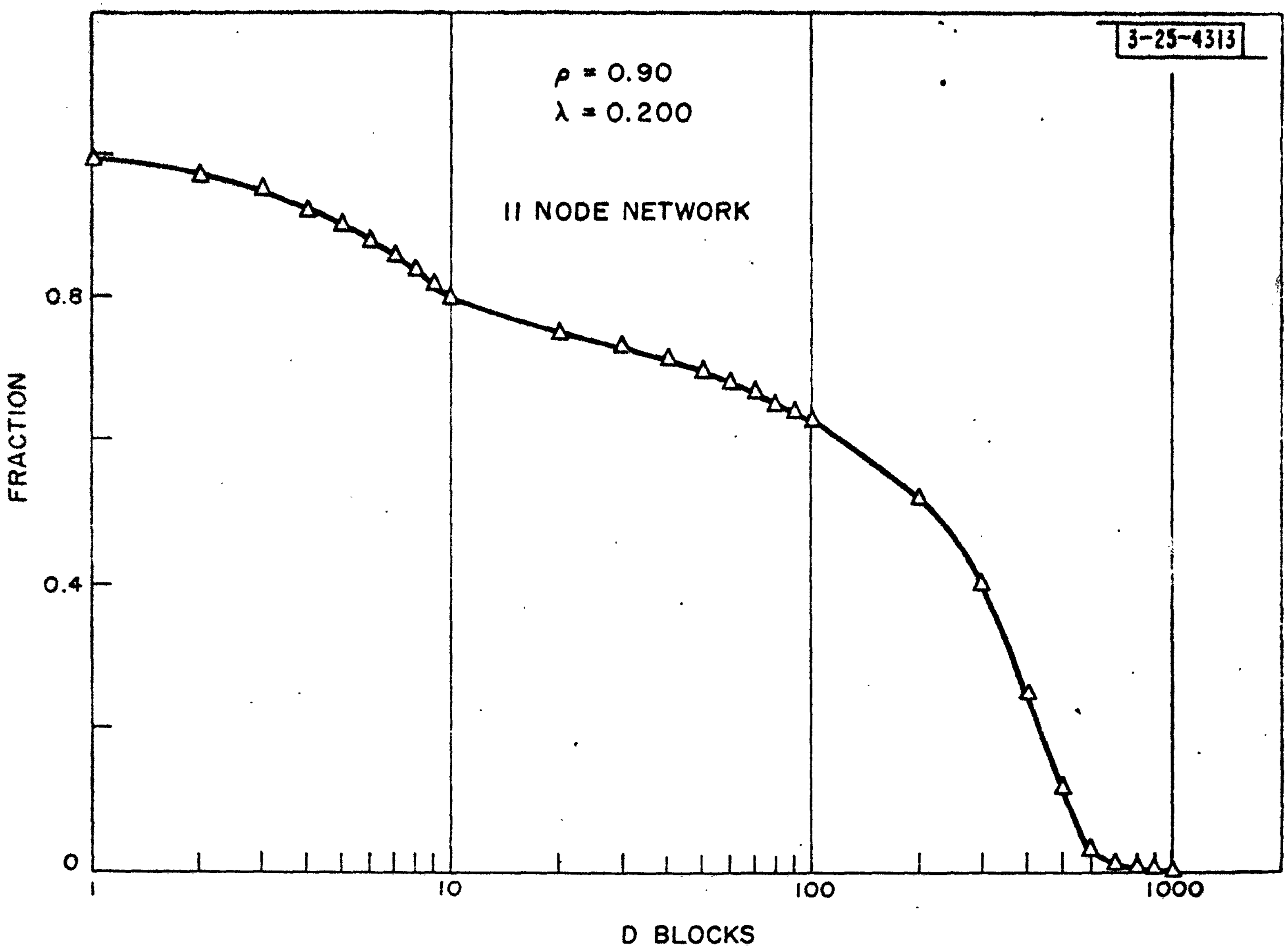

Fig. 21 Fraction of Delays Greater than D Blocks Duration

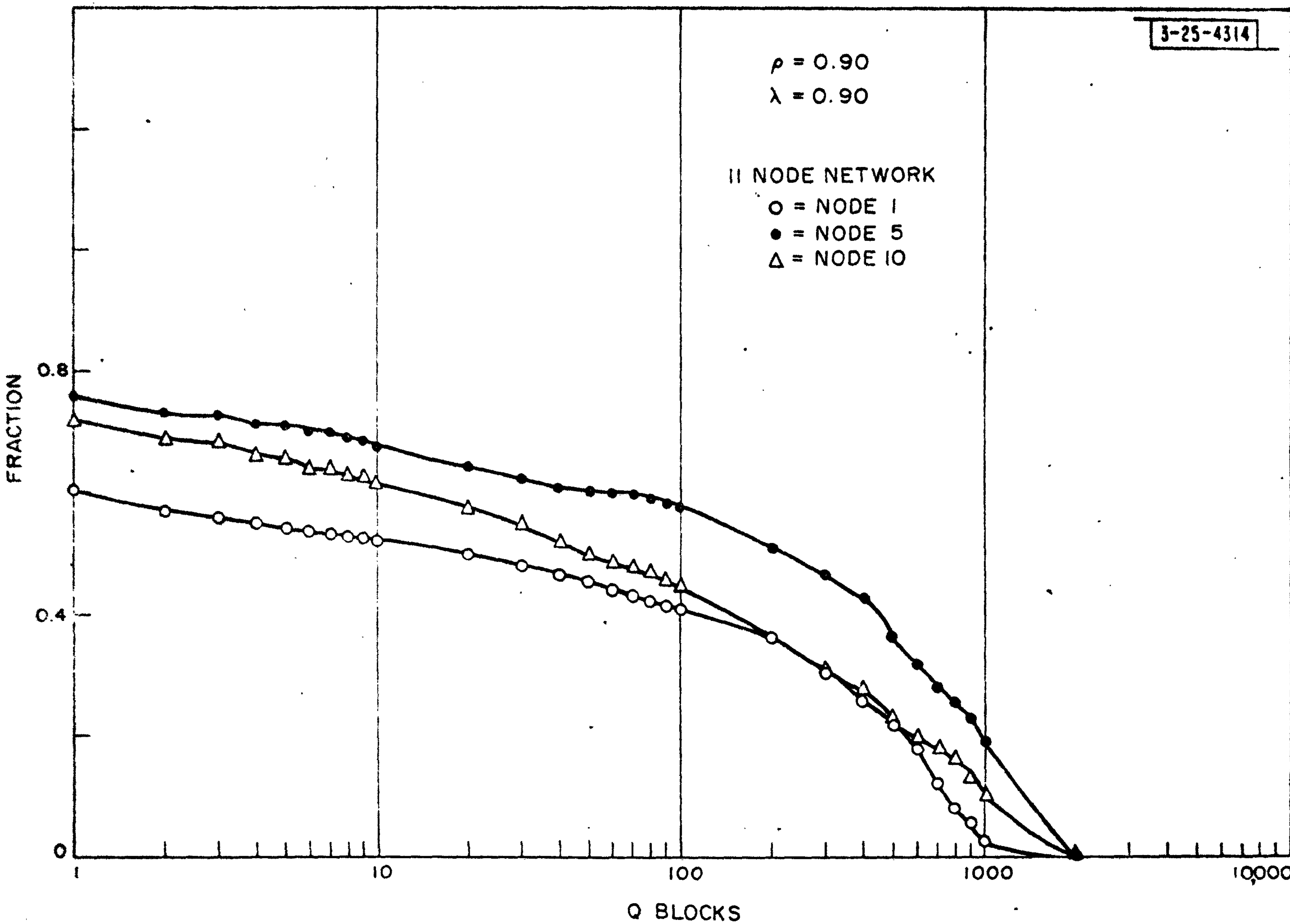

Fig. 22 Fraction of Queues Containing More than Q Blocks

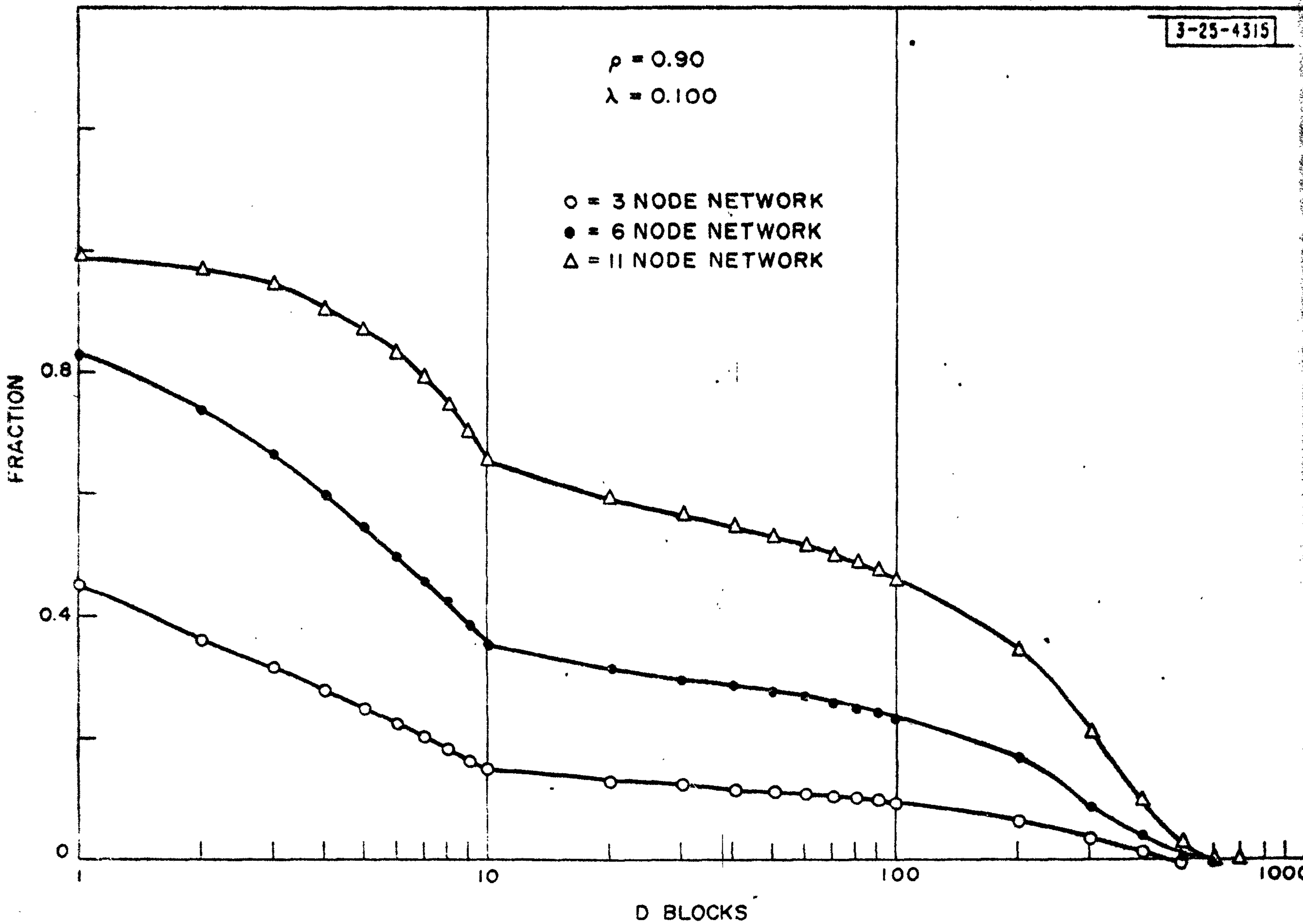

Fig. 23 Fraction of Delays Greater than D Blocks Duration

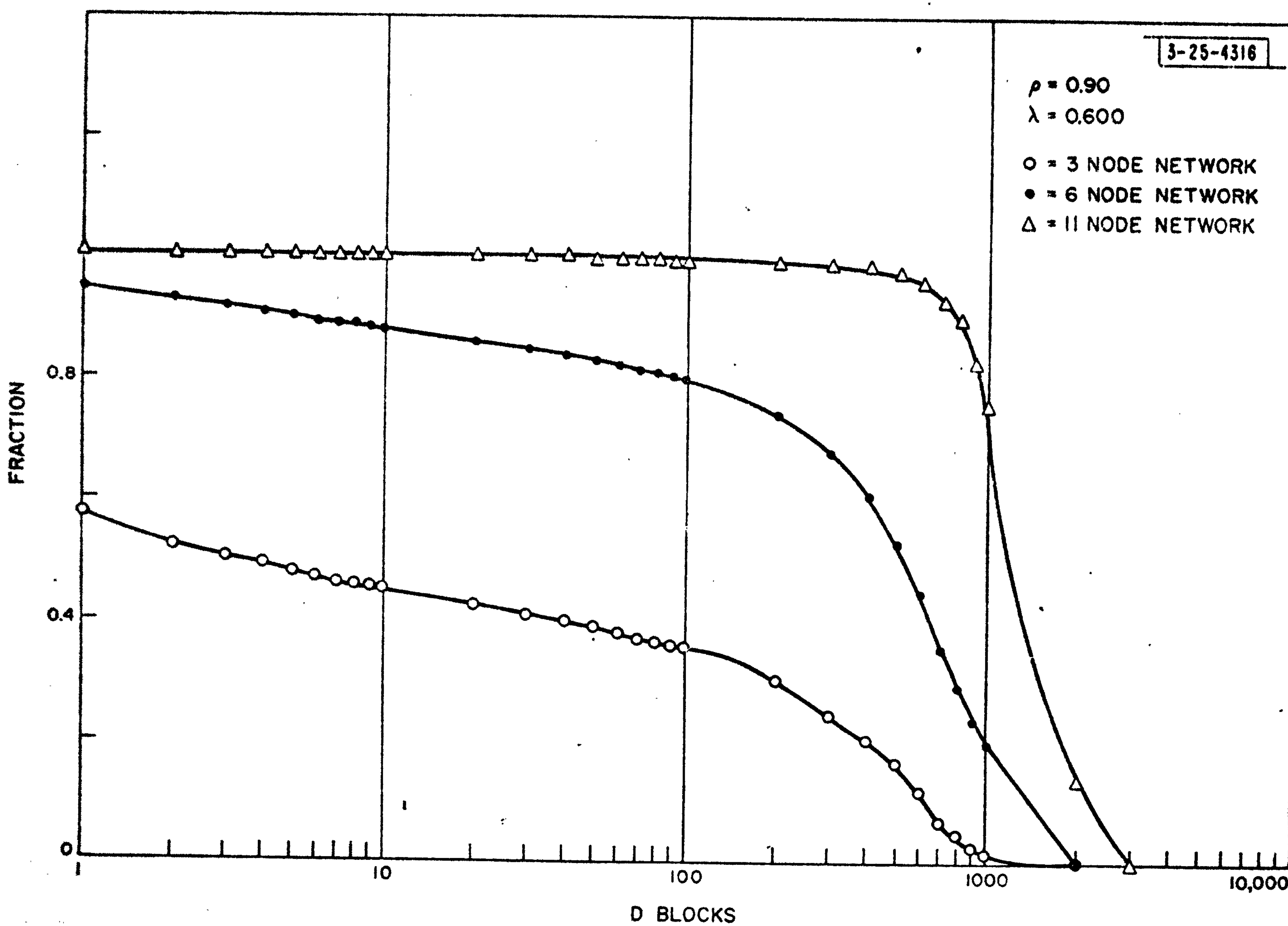

Fig. 24 Fraction of Delays Greater than D Blocks Duration

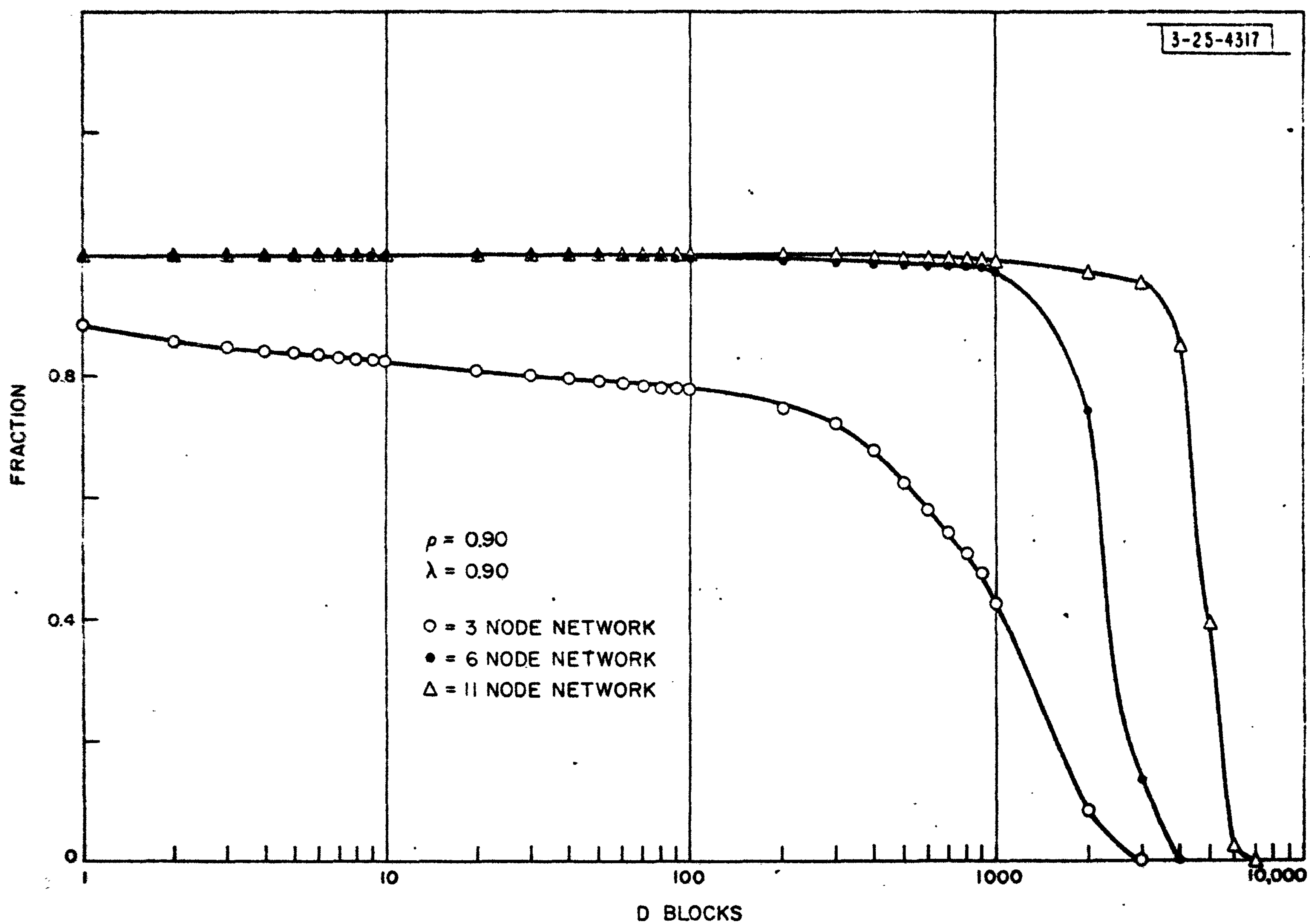

Fig. 25 Fraction of Delays Greater than D Blocks Duration

APPENDIX A

Error-Correction by Requested Retransmissions

In the go-back-two method outlined in Ref. 1, an outage in either channel effectively shuts off communication in the other channel. In this method a slight alteration in system logic decouples the channels, thus allowing data transmission in one channel when the other is faulty. This results in increased data-handling capacity. Assume the feedback path is temporarily out. During this time the source has no information as to whether the blocks it has transmitted have been correctly received by the sink, and it therefore simultaneously stores these blocks for re-transmission or erasure as determined when the feedback path again clears and the sink advises of the status of received blocks.

During the outage, the sink also stores the incoming data with gaps where erroneous blocks occurred. There is thus a one-to-one correspondence between the blocks stored at the sink and the source, and no ambiguity will arise when the sink requests the retransmission of (say) the third word. The sink may specify the blocks received erroneously, either individually or where successive blocks were in error, by the first and last block in this sequence. If more than a certain number of blocks were received in error, the rules may be modified so that all the blocks would be retransmitted.

APPENDIX B

Channel Behavior

For the Bermuda link two frequencies, 5.220 Mcps (night frequency) and 10.385 Mcps (day frequency), were used and about 300 hours of data were obtained. For the Johannesburg link four frequencies between 9 Mcps and 20 Mcps were used and approximately 160 hours of data were obtained.

A single teletype character was repeatedly transmitted and compared at the receiver with the same character locally generated. The time of occurrence of bits in error was recorded on paper tape. The paper tape data was processed on a digital computer to obtain the behavior of the channel assuming code blocks of various lengths had been used. The output of this program was a sequence of positive (a run of correct blocks) and negative (a run of erroneous blocks) numbers.

For the present report a block length of 255 bits was used. The total number of resulting blocks for the Bermuda links was 122,000 and for the Johannesburg link 129,000.

The following procedure was observed for recording data:

1. If three or less successive characters contained bits in error, then the exact bit-error pattern was recorded.
2. If more than three successive characters contained bits in error, the exact bit-error pattern of the first three

characters was recorded, but only the number of the remaining characters in error in this sequence was recorded (the "excess error" count).

The program for obtaining run lengths assumed that these erroneous characters occurred consecutively immediately after the last character whose bit error pattern has been recorded. It was also assumed that the bit-error pattern in these erroneous characters was identical with that of the last character whose bit pattern has been recorded.

3. If more than 384 successive characters contained errors, the recording equipment was automatically turned off and could only be turned on again by the test personnel.

In summary, the recording procedures destroyed some details of the channel behavior, and it is open to conjecture how accurate are the methods of restoring this detail.

REFERENCES

1. B. Reiffen, W. G. Schmidt, and H. L. Yudkin, "The Design of an Error-Free Data Transmission System for Telephone Circuits," AIEE Transactions Committee and Electronics, No. 55, pp. 224-231 (July 1961).

2. A. B. Fontaine, "Applicability of Coding to Radio Teletype Channels," 25G-3, [U], Lincoln Laboratory, M.I.T. (October 1961).

3. B. J. Moriarty, "Blocks in Error and Pareto's Law," 25G-15, [U], Lincoln Laboratory, M.I.T. (January 1963). Related to a study on telephone circuits by J. M. Berger, ASDD, and B. Mandelbrot IBM Research Center and Harvard University, "A New Model for the Clustering of Errors in Telephone Circuits."